THREE ESSAYS ON PROBLEM-SOLVING IN COLLABORATIVE OPEN PRODUCTIONS

by

Marco Tonellato

A dissertation submitted for the degree of

Doctor of Philosophy (PhD) in Economics

Institute of Management

Faculty of Economics

Università della Svizzera italiana (USI), Lugano, Switzerland

Committee Members:

Prof. Alessandro Lomi (thesis advisor, Università della Svizzera italiana)

Prof. Nickolaus Beck (Università della Svizzera italiana)

Prof. Martin Kilduff (University College London)

Prof. Andrew Parker (Grenoble Ecole de Management)

Prof. Christian Steglich (University of Groningen)

September 2014

To my parents

ABSTRACT

The term “open production” is frequently used to describe production systems that rely on volunteer participants who are willing to participate, produce, and bear private costs in order to provide a public good. Examples of open production are becoming increasingly common in many industries. What make these productions possible? How may they be sustained in a world of organizations in which the evolutionary products of economic selection are elaborate hierarchical forms of organization? One way to address these questions is to look at how open productions solve problems that are common to all production organizations such as, for example, problems in the division of labor, allocation of tasks, collaboration, coordination, and maintaining balance between inducement and contributions. Under the conditions of extreme decentralization that are the defining feature of open productions, this approach implies a detailed observation of individual problem solving practices. This is the approach I develop in my dissertation. Unlike much of the prior literature on open productions, I deemphasize motivational elements, status-seeking motives, and allocation of property rights issues. I focus instead on actual work practices as revealed by the day-by-day problem solving activities that qualify open productions projects as production organizations despite the absence of formal contractual arrangements to regulate principal-agent relations. What my work adds to the extensive, informative, and well-developed discipline-based explanations that are currently available, is a focus on the emergence of micro-organizational mechanisms through which problem assignment (Chapter 2), problem resolution (Chapter 3), and sustained participation (Chapter 4) are obtained in open productions. In my essays, I draw from organizational sociology and the behavioral theory of the firm to specify models that relate individual problem-solving activities to structured patterns of action through emergent work practices. In the models that I specify and test, I emphasize processes of attention allocation (Chapter 2), repeated collaboration and group diversity (Chapter 3) and identity construction (Chapter 4) as central to our understanding of the dynamics of problem-solving in organizations. One element of novelty in my study is that my research design makes these work practices directly observable at a level of detail, completeness, and precision that was inaccessible in the past. To illustrate the empirical value of the view that I develop I examine problem-solving activities – i.e., bug fixing and code production – within two Free/Open Source Software (F/OSS) projects during their entire life span. Readers of my work will know more about how organizational micro-mechanisms emerge in open productions.

TABLE OF CONTENTS

ABSTRACT.....v
TABLE OF CONTENTS.....vii
LIST OF FIGURES.....ix
LIST OF TABLES.....x
AKNOWLEDGEMENTS.....xi

CHAPTER 1: INTRODUCTION.....1
1.1 Motivating Question.....2
1.2 Research On Collaborative Open Production.....6
1.3 Open Productions As Problem-Solving Organizations.....10
1.4 Outline Of The Empirical Studies.....11
1.5 Empirical Setting.....16
1.6 Outline Of The Thesis.....18

CHAPTER 2: RELATIONAL COORDINATION IN AN OPEN SOURCE SOFTWARE PROJECT: THE ROLE OF ATTENTION NETWORKS.....24
2.1 Introduction and motivation.....25
2.2 Illustration of research issues in the empirical context.....29
2.3 Theory and hypotheses.....32
2.3.1 Attention clustering.....36
2.3.2 Attention spread.....39
2.3.3 Interaction hypotheses: the role of individual experience.....41
2.4 Methods.....42
2.4.1 Data.....42
2.4.2 Variables and Measures.....43
2.4.3 Stochastic Actor-oriented models.....50
2.4.4 Model estimation and evaluation.....52
2.5 Results.....53
2.6 Discussion and conclusions.....57

CHAPTER 3: THE EFFECT OF EXPERTISE DIVERSITY ON GROUP LEARNING AND PERFORMANCE: A CASE STUDY IN OPEN SOURCE SOFTWARE.....67
3.1 Introduction.....68
3.2 Theory and Hypotheses.....71
3.3 Empirical Setting: Bug-Fixing in Free/Open Source Software Projects.....79
3.4 Methods.....82
3.4.1 Data and Sample.....82
3.4.2 Measures and Variables.....85

3.4.3 Model specification and estimation....91
3.5 Results....93
3.6 Discussion and Conclusions....99

CHAPTER 4: IDENTITY CONSTRUCTION AND SUSTAINED PARTICIPATION IN AN OPEN SOURCE SOFTWARE PROJECT....109
4.1 Introduction and Motivation....110
4.2 Theory and Hypotheses....113
4.2.1 Literature review: participation in open productions....113
4.2.2 Identity construction in open productions....116
4.3 Empirical setting: code development in the Apache HTTP server....121
4.4 Methods....123
4.4.1 Data and sample....123
4.4.2 Variables and measures....125
4.4.3 Empirical model specification and estimation....130
4.5 Results....131
4.6 Discussion and Conclusions....140

CHAPTER 5: CONCLUSIONS....147
5.1 Summaries of chapter results and contributions....148
5.1.1 Chapter 2....149
5.1.2 Chapter 3....150
5.1.3 Chapter 4....152
5.2 Scope conditions and limitations....153
5.2.1 Modularity....155
5.2.2 Transparency....156
5.2.3 Technology-mediated Communication....157

LIST OF FIGURES

Fig. 1.1 Inputs-Mediators-Outputs-Inputs Model.....11
Fig. 1.2 Constructs studied in the empirical part of the dissertation and their relations.....12
Fig. 2.1 The observed organizational attention network.....30
Fig. 2.2 Attention Clustering visual representation.....38
Fig. 2.3 Attention Spread visual representation.....39
Fig. 3.1 Repeated Collaboration visual representation.....87
Fig. 3.2 Expertise Diversity visual representation.....89
Fig. 3.3 Kaplan-Meyer estimator for bug resolution in Apache HTTPD Server, 2001-2013.....94
Fig. 3.4 The relationship between repeated collaboration, expertise diversity and bug resolution in Apache HTTPD Server, 2001-2013.....98
Fig. 4.1 Kaplan-Meyer estimator for time of sustained participation, Apache HTTP Server, 1996-2013.....134
Fig. 4.2 Interaction between Cumulated Experience and Identity Specialization.....138
Fig. 4.3 Interaction between Repeated Collaboration and Identity Specialization.....139

LIST OF TABLES

Tab. 2.1 Summary of Attentional Mechanisms of Theoretical Interest.....46
Tab. 2.2 Summary of Endogenous Network Controls.....47
Tab. 2.3 Summary Table for Exogenous Actor-specific Covariates.....48
Tab. 2.4 Summary of Actor-relations Interaction Terms.....49
Tab. 2.5 Method of Moment Estimates of SAOM for Bipartite Networks.....54
Tab. 3.1 Descriptives and Correlations Table for Bug Resolution in Apache HTTPD Server, 2001-2013.....93
Tab. 3.2 Repeated events Cox hazard regression of bug resolution in Apache HTTPD Server, 2001-2013.....95
Tab. 4.1 Descriptives and correlations table for code commits to Apache HTTPD Server, 1996-2013.....133
Tab. 4.3 Cox hazard regression of code commits to Apache HTTPD Server, 1996-2013.....135

AKNOWLEDGEMENTS

The incredible, life-changing journey that brought me to this final stage of my tenure as PhD student has been challenging, maybe hard and frustrating at times, but extremely rewarding overall. Needless to say, I wouldn't be here now if it wasn't for the invaluable encouragement of many people whom I had the privilege to have around me during this long ride.

First of all I would like to thank my advisor Alessandro Lomi, whose constant support and outstanding intellectual stimuli have been paramount for my formation as a researcher. He has been a fantastic mentor who worked relentlessly to provide me with all I needed – and much more – to pursue my ambitions of becoming a management scholar. Ever since he recruited me, he has opened up so many doors for me, and believed in me when I had doubts. I most sincerely hope that we will continue collaborating on new, exciting projects in the future.

I owe a special debt of gratitude to my friend and colleague Guido, without whom this thesis wouldn't have been possible. He shared with me his passion for research on open source communities, which inspired so profoundly my intellectual growth. He helped me immensely in collecting, coding and managing the data that were used in this dissertation. He has been a terrific counselor and source of relief when things weren't the best. Special thanks to Francesca for her invaluable intellectual as well as emotional support, to Vanina for her precious advices and feedback during seminars and conferences, to Beth for her crucial and passionate help and to the rest of Alessandro's collaborators in Lugano, Alberto, Christoph, Giorgia and Paola.

I would also like to express my gratitude to all students, post-docs and faculty members at the Institute of Management in Lugano. At IMA I found the best learning environment I could hope for, and I've benefited from an endless number of fruitful discussions that helped me craft, challenge and develop the ideas on which this dissertation is based. In particular I would like to thank Chanchal, Ivona, Jaime, Margarita, Martina and Pooya, all of whom have been fantastic colleagues in the doctoral program. I feel privileged for the opportunity to share ideas, gossip, worries, and drinks with them.

I am also extremely grateful for the time spent at Carnegie Mellon University where I had the enormous privilege to have the chance to discuss my work with some of the smartest and nicest scholars in the world. Special thanks to faculty members Linda Argote and Brandy Aven for their

terrific mentoring and to PhD students Jonathan Kush and Evelyn Zhang for our endless discussions, brainstorming and sharing of research ideas.

I would like to thank my committee members Nickolaus Beck, Martin Kilduff, Andrew Parker and Christian Steglich who kindly accepted the burden of reading through my material and providing me with helpful comments and remarks. Special thanks to Andrew for believing in me and being the main sponsor of my next, exciting career challenge in Grenoble.

Finally I would like to mention the most important persons in my life, my parents Franca and Mauro and my brothers Michele and Giulio, for just being there for me at any time, without the need to ask for anything at all. I feel incredibly blessed to be part of our family. This thesis is dedicated to you, to us.

Last, but not least, thank you Marika. Your generous patience, your continuous support, your love throughout this amazing yet extremely demanding endeavor mean the world to me. You are the best.

CHAPTER 1

INTRODUCTION

1.1 Motivating Questions

"*Is the apparently anarchistic process of open source production, in which no one tells anybody else what to do, a new model of business organization?*" (Josh Lerner and Jean Tirole)

When economists Josh Lerner and Jean Tirole (2001: 821) raised this fundamental question, Free/Open Source Software (F/OSS) was just on the verge to turn from a curiosity for few enthusiasts into a mainstream phenomenon affecting business corporations, end users and policy makers alike (Levine & Prietula, 2014). Now F/OSS is ubiquitous and the number of F/OSS projects is constantly growing. Sourceforge, the most popular F/OSS infrastructure repository, hosts more than 430,000 projects and 3.7 million registered users. Each day users of Sourceforge alone download on average more than 4 million programs, commit more than 14,000 changes to source code, and track more than 2,700 software bugs. The Linux and Android operating systems, and the Apache Web Server are the largest, most successful and influential projects. They operate on millions of computers and devices and rely on the distributed contributions of thousands of individuals. F/OSS projects span a wide range of applications, including programming languages (such as Perl and Python), user software (such as Mozilla Firefox and LibreOffice) and programming environments (such as Eclipse).

The term Free/Open Source Software refers to software released under a license that permits inspection, use, modification and redistribution of the software's source code (Crowston et al., 2012). F/OSS represents an alternative approach to technology development and innovation. Rather than implementing formal intellectual property rights that induce the formation of contracts between firms, developers and final users, F/OSS projects "open" their boundaries by enlisting all participants as project contributors and giving them access to the software source code. F/OSS projects aim at

providing private incentives to maximize adoption and individual contributions but reduce the opportunities for appropriating the economic value of the innovation, as in the case of public goods (West, 2003). F/OSS may be seen as a private-collective innovation model that combines elements of individual investment and collective action (von Hippel & von Krogh, 2003). This innovation model is based on the creation of incentives for individuals and organizations to sustain private costs in order to generate public goods innovations, which are non-rival and non-exclusive in consumption. Although software with F/OSS licenses and proprietary software may be developed in the same way, most F/OSS products are developed by teams of geographically dispersed participants who often work on a voluntary basis (Lee & Cole, 2003).

The private-collective model of collaboration, innovation and production can now be generalized beyond software. The term "open production" has been introduced to describe production systems relying mostly on volunteer participants who are willing to collaborate, bear private costs and defer self-interest in order to provide a public good (Baldwin & von Hippel, 2011). Similarly to Levine and Prietula's (2014) definition of open collaboration, open productions are entities that (i) create goods of economic value; (ii) grant open access to participants to contribute and consume freely; (iii) are based on constant interactions and information exchange; (iv) coordinate purposefully participants' labor. Examples of open production are becoming increasingly common in engineering (Enkel, Gassman & Chesbrough, 2009), health care (Kamel Boulos & Wheeler, 2007; Eysenbach et al., 2004), product design (Jeppesen & Fredericksen, 2006; Dahlander & Wallin, 2006), and software development (Raymond, 1999; West & Gallagher, 2006) – the specific production activity that I examine in my dissertation.

What makes these productions possible, and how may they be sustained in a world of organizations in which selection favors the proliferation of hierarchical forms of organization (Simon,

1965)? How can production organizations "in which no one tells anybody else what to do" (Lerner and Tirole, 2002: p. 821) substitute traditional organizational arrangements such as hierarchy-based decision making and contracts that regulate principal-agent differences in the incentive structure?" These are questions of core theoretical importance in the study of organizations. During the last decade economists (Lerner & Tirole, 2002; 2001), organizational sociologists (Bailey, Leonardi & Barley, 2012), political scientists (Feldman, 2010), students of technological innovation (Baldwin & von Hippel, 2011; Dahlander & Gann, 2010) and legal scholars (Benkler, 2006) have recognized the importance – and difficulty – of providing coherent answers in the context of open production. In my dissertation I address these difficulties directly, by examining the internal organizational logic of open production – i.e., by considering open production projects as specific forms of economic production. As such, open productions have to address the same issues facing conventional hierarchical organizations such as, for example, division of labor, allocation of skills to tasks, collaboration, and coordination. My work adds to existing research a focus on the micro-organizational mechanisms that make open production effectively possible as a strategy for technology development. Unlike much of the prior literature on open productions, I do not emphasize incentives, status-seeking or efficient allocation of property rights. I concentrate instead on the actual day-to-day activities through which fundamental organizational mechanisms such as task assignment, collaboration for problem resolution, and sustained participation emerge and ultimately make open productions sustainable problem-solving arrangements. The research design that I have implemented and the models I have developed allow me to examine work practices at a level of resolution that has so far been inaccessible for prior research.

I focus on individual acts of problem solving - the smallest possible constituent unit of work practices, emphasizing emergence of micro-organizational mechanisms through which problem assignment (Chapter 2), problem resolution (Chapter 3), and sustained participation (Chapter 4) are obtained in open productions. In particular, I am interested in understanding how interaction between

participants and problems generates and sustain the basic coordination micro-mechanisms produced by hierarchical arrangements in more conventional production organization. Inspired by this general orienting question, I aim at answering the following research questions, each centered on a specific aspect of such emergence. Highlighting interdependencies of attention allocation processes, in Chapter 2 I ask "How do individuals in self-managing, decentralized work teams decide which task to work on"? In Chapters 3 and 4 I instead focus more on the performance implications of collaborative work practices in open productions, in terms of time to solve problems and the likelihood to retain and sustain voluntary contributions in presence of fluid organizational boundaries. Stressing processes of repeated collaboration and expertise heterogeneity in groups, in Chapter 3 I ask: "How does the internal composition of collaborative work groups in open productions affect the time in which problems are solved"? Finally, based on the idea that voluntary contributors construct their identity profiles as specialists or generalists by narrowly focusing or widely dispersing their efforts across knowledge domains, chapter 4 addresses the following question: "How do decentralized work groups in open productions manage to retain their human capital despite the presence of porous boundaries that favor constant member turnover"? In the remaining parts of the introduction I expand and motivate these research questions in a broader context informed by organizational sociology, the behavioral theory of the firm and open innovation.

The goals of this dissertation may be summarized as follows: the first goal is to provide an overarching framework for the concept of "open production", showing how it emerges from, and extends beyond, the basic model of problem-solving organizations. I build on previous work on open innovation (Lakhani, Lifshitz-Assaf & Tushman, 2013) and decentralized production communities (Benkler & Nissenbaum, 2006), but capture a broader range of processes, drawing from a behavioral perspective on organizations (Gavetti et al., 2012), including very fundamental concepts such as attention allocation, organizational learning, and division of labor. The second goal is to extend current

explanations based on motivation and incentives to show how open productions are sustained through mundane work practices. In open productions problem-solving acts that link individuals to tasks are transparent (i.e., immediately visible) to all participants. I argue that both individual decision-making and organizational outcomes are embedded in, and determined by, this transparent problem-solving structure. The third goal is to identify future avenues of research related to open productions to inspire organizational scholars to study how these new organizational arrangements affect collaborative practices and the production of innovation.

1.2 Research on collaborative open production

In the last fifteen years, organizational scholars have tried to understand how open production in general, and F/OSS projects in particular, work as organizational solutions to the problem of producing and distributing public goods (Bonaccorsi & Rossi, 2003). The task has proven particularly challenging because of the ‘‘anarchistic’’ nature of the production process of F/OSS. This type of “open production” manifests itself in extreme forms of decentralized decision-making, and in the absence of exogenously established hierarchical structures and formal coordination mechanisms (Lerner & Tirole, 2001). How are F/OSS productions possible under such apparently adverse organizational conditions? More specifically I ask how do fundamental organizational mechanisms such as the ones outlined in the previous paragraph emerge in these conditions to make open productions sustainable. Available research suggests a variety of possible answers such as the expected returns on reputation and knowledge that participation in successful projects may afford (Lakhani & von Hippel, 2003; von Krogh & von Hippel, 2006), the presence of a diffuse shared gift-exchange culture (Bergquist & Ljungberg, 2001; Zeitlyn, 2003), the widespread availability of efficient and effective communication technologies (Lanzara & Morner, 2005; Scacchi, 2004), the fact that direct monitoring may act as an efficient safeguard against free riding problems (Baldwin & Clark, 2006),

and the presence of conscious processes of boundary maintenance (Ferraro & O'Mahony, 2012). Available explanations typically strive to answer the fundamental question, first raised by Lerner and Tirole in their seminal paper (2002: 198): "Why would thousands of top-notch software developers contribute for free to the creation of a public good?" Most research has focused on individual motivations. Empirical studies on this subject have documented a series of factors that induce individuals to contribute to F/OSS projects.

In a recent review Crowston and colleagues (2012) show that motivations are highly heterogeneous and generally fall into three broad families. The first includes extrinsic motivations, which such as status or reputation which may represent signals of competence and hence contribute to future career development (Hars & Ou, 2002; Hertel, Niedner & Hermann, 2003; Stewart, 2005). The second includes intrinsic motivations, like pure enjoyment and satisfaction deriving from sharing information or learning opportunities (Ghosh, 1998; Shah, 2006; Stewart & Gosain, 2006). Finally, the third family includes motivations that are extrinsic in principle, but could be internalized so that they are felt as self-regulating behavior instead of exogenous impositions (Deci and Ryan, 1987; Roberts, Hann & Slaughter, 2006). These internalized extrinsic motivations include, among others, peer recognition and reciprocity. For instance, Lakhani and von Hippel (2003) show that mundane, but necessary tasks in F/OSS projects – tasks that couldn't be explained by purely extrinsic or intrinsic motivations – are performed because participants feel part of a community where free user-to-user assistance is the standard *modus operandi*. Socialization and collaborative work practices induce the establishment of unwritten norms and informal institutions that encourage and foster individual participation in the project. While early work on motivation focused mostly on the intrinsic vs extrinsic framework, recent developments have advanced this latter line of research by exploring the role of actual work practices in sustaining collaborative behavior (von Krogh et al., 2012). Building on the social philosophy of Alasdair MacIntyre (1981; 1998), von Krogh and colleagues (2012) illustrate a

"motivation-practice" framework that depicts how the relationship between individual motivation and economic outcomes is embedded in day-to-day social practices and their supporting institutions.

Although recent studies have documented increasing synergies between business corporations – such as IBM and Microsoft – and open source communities (Dahlander & Magnusson, 2008; West & Gallagher, 2006) this work purposely look at projects composed almost entirely by volunteers, in order to rule out alternative explanations due to extrinsic pay-based motivations. In this dissertation I take an alternative approach grounded in a detailed examination of how individual problem-solving attempts actually happen as a consequence of dynamic work practices linking problems and participants (Pentland, Hærem & Hillison, 2011; McGrath & Argote. 2001). Attention to work practices is not completely new in studies of open productions. In F/OSS projects the actual activities of software contributors have been directly investigated with a focus on their emergent, decentralized character (Crowston, Li, Wei, Eseryel, & Howison, 2007; Crowston & Scozzi, 2008; see also Crowston, Wei, Howison, & Wiggins, 2012, for recent literature reviews discussing this stream of research). For instance, reuse of software code has been investigated as a specific form of knowledge recombination fostering the ability to generate innovation in F/OSS production (Haefliger, von Krogh & Spaeth, 2008). What I add to these studies is an attempt to analyze work practices founded on an overarching perspective on individuals and problems in organizations which views them as existing in a relation of mutual constitution, in the sense of Breiger (1974, 2000, 2002) and Breiger and Mohr (2004). According to this perspective, organizations may be understood in terms of a dual association between individual carriers of potential solutions (whose organizational identities are defined in terms of the problems with which they are associated) and problems (whose organizational identities are defined in terms of the individuals engaged in their resolution). Although important in any organization, the quality of contributors and the nature of their tasks and work practices are rarely sufficient, in isolation, to explain how and why one specific organizational arrangement is more or less effective than the

feasible alternatives. Given the key organizational characteristics of open productions – the weakness of centralized control and formal coordination and the direct and unrestricted access of participants to problems – all the basic components of organizational structure are endogenous, interdependent, and regulated by processes of self-selection and self-assignment (von Krogh, Spaeth & Lakhani, 2003). In particular, self-assignment to problem-solving activities lies at the heart of any F/OSS project because both the implementation of new software functionalities and also ‘‘bug-fixing’’ – that is, the correction of defects and misbehaviors generically called ‘‘software bugs’’ – can be conceived of as problems that contributors have to resolve if their software product is to progress into any sort of stable state.

This leads to a view of open productions as problem-solving organizations in which the micro-mechanisms of problem solving become the focus of investigation. Work practices, observed for the first time at the most fine-grained level of detail, emerge from sequences of actions linking participants and problems in F/OSS projects. As Lakhani and von Hippel write (2003: p. 940): “We think that it is important to analyze the micro-level functioning of successful open source projects to really understand how and why they work. For example, we think it would be useful to conduct empirical studies to explore other puzzling aspects of how an open source project functions such as: how is coordination achieved among open source software contributors; how can problems be segmented into modules of a size that fit the sources and incentives of individual users to effectively contribute?” Although these suggestions were put forward over ten years ago, most literature on F/OSS and open innovation either investigated a variety of motivation and incentives-related reasons for understanding contributors’ participation or explored a series of hybrid strategies for firms to tap into open production communities to produce innovation. Those questions may be addressed only by considering open productions as specific problem-solving arrangements and by concentrating on their internal organizational logic. My research design and the very fine-grained level of detail of the continuous time problem-solving data that I collected allow for one of the first times to respond to that call.

1.3 Open productions as problem-solving organizations

One way to think about organizations is as problem-solving arrangements designed to economize on the limited cognitive resources of their members, simplify decisions, encourage coordination, and reproduce production relations (Cyert & March, 1963; March & Simon, 1958). Building conceptually on the path breaking work of Herbert Simon on problem-solving (Simon, 1969; Newell & Simon, 1972) and on its more recent "evolutionary" developments (Marengo et al., 2000; Marengo & Dosi, 2005), I consider organizational problem-solving as a particular form of production activity. I posit that the dual association linking participants to problems generates information through individual acts of production. Participants use this information to allocate their attention and decide their level of engagement with the overall project. The global structure of this open production system emerges from individual acts of problem-solving which crystallize into interdependent work practices – defined analytically as structured sequences of problem-solving events. The individual decision to engage organizational problems is modeled as a function of: (i) the characteristics of the problems; (ii) the characteristics of the participants, and (iii) the characteristics of the associations linking participants to problems.

In this dissertation I consider "bug-fixing" – or attempts by project participants to resolve software problems (software "bugs") – and production of software code as the core organizational problem-solving activity. Bug-fixing possibly represents the most transparent example of organizational problem-solving, as it exemplifies a setting in which problems, problem-solvers and solutions encounter each other in a space of decision opportunities (Cohen, March & Olsen 1972). Bug-fixing and code production have long been recognized as the essential activities in the development and maintenance of F/OSS products (Crowston, 2008; Michlmayr, 2007) and are frequently viewed as providing a significant contribution to the success of F/OSS projects (Crowston &

Scozzi, 2008; Crowston et al., 2012). Yet, the conditions of extreme decentralization characterizing problem-solving activities in F/OSS environments poses the question of how collaboration may be fostered and sustained in the absence of centralized control. In decentralized productions characterized by distributed work carried out by highly autonomous participants, the answer to this question does not reside in the design of optimal hierarchical control systems or incentives. My attempt to address this concern starts from a simple representation of organizational problem-solving activities as embedded in structures of connections linking problems and participants – the latter being the bearers of potential solutions (Cohen March & Olsen, 1972).

1.4 Outline of the empirical studies

I organize the empirical chapters of this work using the inputs-mediators-outputs-inputs (IMOI) model (Ilgen, et al., 2005) – a general framework that has been repeatedly used in research on teams and teamwork (Hackman & Morris, 1978; McGrath, 1991). Figure 1 shows the baseline IMOI model.

Figure 1.1: Inputs-Mediators-Outputs-Inputs Model (adapted from Ilgen et al., 2005)

This model most closely matches the theoretical constructs I use to support my arguments and provides additional structure for framing the empirical studies of my dissertation. I opt for the IMOI model rather than its predecessor, the Input-Process-Output (IPO) model (Hackman & Morris, 1978) because it postulates the existence of feedback processes linking outputs to inputs, conceiving outputs as inputs to upcoming processes (Crowston et al., 2012). This view is consistent with the idea that the constant encounter of participants and problems in a transparent environment endogenously generates new solutions and new opportunities for problem-solving. Furthermore, this framework is consistent

with the idea that while organizations transform inputs into outputs they are also transformed by the very same process (Padgett & Powell, 2012; Padgett, Lee & Collier, 2003). This model of team production is also apposite because problem-solving processes in open production occur in small teams with porous boundaries whereas most studies have been conducted at the project level of analysis. When needed I adapt the framework to include theoretical constructs directly related to the development of F/OSS software.

Figure 1.2 reports the resultant framework with the relevant constructs that I adopt in the empirical part of the dissertation. Inputs characterize starting conditions of the production process, such as organizational participants' characteristics and problems characteristics. Mediators characterize processes that drive the transformation of inputs into outputs. These processes represent typically group level variables or dynamic interdependencies among individuals that affect organizational participants as they work to solve their problems, conducting to the outputs of the model. Examples of mediators are social and cognitive processes that underlie the production process, such as collaboration, allocation of attention and decision-making. They usually represent the variables of theoretical interest in the models I specify. Outputs characterize relevant consequences of the production process, typically performance measures such as time to fix problems or quantity of innovation produced. They are usually, but not always, the dependent variables in my models.

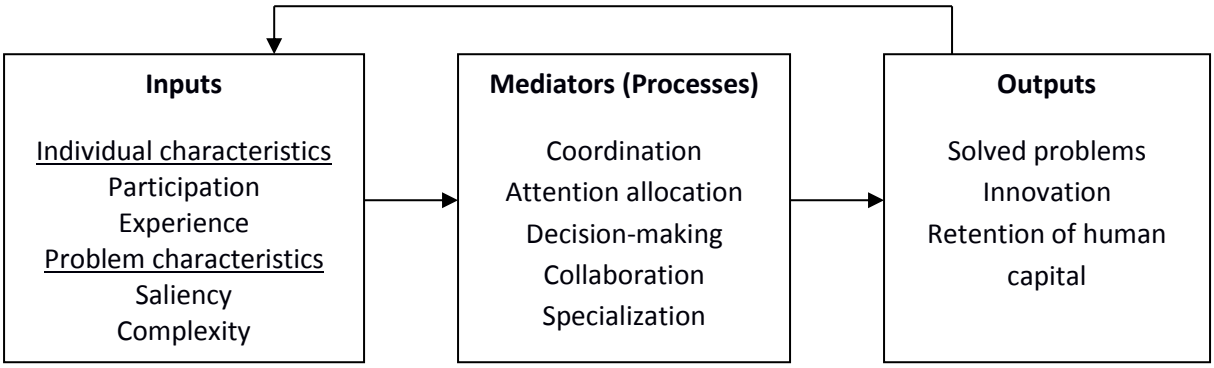


Figure 1.2: Constructs studied in the empirical part of the dissertation and their relations

I build on behavioral theories of organizations to specify models that relate individual problem-solving activities to the structured pattern of action through emergent work practices (Barley & Kunda, 2010). In the models that I specify and test, I emphasize processes of relational coordination – i.e., coordination emerging from direct interaction among participants (Gittell, 2002) – as central to our understanding of the dynamics of problem-solving in open production. Not only do I build on behavioral theories of organizations, but also I maintain the analytical focus on the interaction among individuals determined by joint involvement in problem-solving activities. To illustrate the empirical value of this perspective I examine problem-solving activities within several large F/OSS projects. This setting is particularly useful because the decentralization typical of F/OSS projects implies that organizational structures are not exogenously imposed, but emerge directly from the problem-solving activities of individual participants. This empirical setting facilitates more direct observation of the emergence of organizational structures and routines from actual work practices linking participants to problems. More specifically, I focus on processes of attention allocation (Chapter 2), repeated collaboration and expertise diversity (Chapter 3) and identity specialization (Chapter 4) which take place when a distributed group of software contributors coordinate to produce software and fix bugs.

Figure 1.2 illustrates the overall organization of my work into three studies – each examining specific aspects of problem solving in collaborative open productions. The first study (Chapter 2) is designed to address the following question: "How do individuals in decentralized work teams decide which task to work on"? The chapter focuses on inputs and processes to explain the mechanisms behind task self-assignment decisions. The chapter builds on the view that organizational decision-making is the outcome of interacting flows of problems, opportunities and problem solvers (Cohen, March & Olsen, 1972). The study is centered on the idea that attention allocation is the mechanism by which problems and problem solvers meet each other in a space of decision opportunities (Ocasio, 2012). Because attention is a scarce resource in information-intensive organizations, understanding

patterns of attention allocation is crucial for achieving effective coordination and division of labor (Simon, 1947; Sullivan, 2010). Extant literature has focused mainly on bottom-up (i.e., driven by problem characteristics) or top-down (i.e., driven by problem solvers' cognitive schemas) processes of attention allocation in organizations. I argue that problem selection decisions are embedded in a relational context that constitutes patterns of information usage, inducing organizational members to lower their search costs and better evaluate problems at stake. Hence, problem selection decisions don't occur in isolation, but are situated within a connected system made of attention allocation ties linking problems to other individuals. I analyze the structure of this system by proposing and investigating two fundamental mechanisms that sustain a situated view of attention: attention clustering and attention spread. By using newly developed Stochastic Actor Oriented Models (SAOM) for bipartite graphs (Conaldi, Lomi & Tonellato, 2012), I find that problem selection decisions show a tendency towards attention clustering and a tendency toward disassortative attention spread, and that these two effects are compounded by the high levels of experience of problem solvers. This chapter contributes to the literature on organizational attention and distributed cognition by extending recent work on attention networks (Prato and Stark, 2013). Whereas current work emphasizes that market valuations are embedded in attention networks that shape individual perceptions, my study sheds light on two specific mechanisms that inform organizational participants when they self-select specific problems to solve.

The second and third studies (chapters 3 and 4) focus more on processes and outputs, as they intend to illuminate the factors enhancing performance in open productions. In the second study I delve into the performance implications of distributed problem-solving in open productions by looking at how organizations learn to coordinate activities that reduce the average problem latency time, or the time passed between the appearance of a problem and its resolution. The chapter addresses the following question: "How does the internal composition of decentralized work groups in open productions affect the way in which individuals collaborate to solve problems"? The emergence of

open, collaborative forms of economic organization, such as open source software projects, invites reconsideration of the relationship between organizational structures and learning activities. Repeated collaboration represents a crucial mechanism through which organizations learn. Research on group learning shows that experience in working together helps groups within organizations to achieve better coordination, establish shared norms, and reduce task completion time (Reagans, Argote & Brooks, 2005). However, extended collaboration time may induce group members to overly focus on existing solutions and available routines, which in turn hinders the search for solutions that may be more effective but more distant from established practices (Katz, 1982; Levinthal & March, 1993). Following existing work (e.g., van der Vegt & Bunderson, 2005) I argue that group learning is contingent on the distribution of task-related expertise among group members. My results build on prior research and go beyond it by suggesting that the effect of repeated collaboration on group performance is moderated by the degree of expertise diversity in the group. Groups that are composed of individuals with homogeneous expertise outperform groups with heterogeneous expertise at lower levels of prior collaboration, because homogeneity helps establish common ground in the early phases of collaborative practices. At the other end, groups composed of individuals with heterogeneous expertise outperform groups with homogeneous expertise at higher levels of prior collaboration, because specialization and expertise diversity are essential to establish a clear understanding of "who knows what" in the group (Liang, Moreland & Argote, 1995). This study is important as it extends theory on organizational learning by investigating relevant contingency effects for performance-based learning, whereas prior work has instead supported arguments about unconditional and linear effects of experience working together.

The third study addresses the following question: "How do decentralized work groups in open productions manage to retain their human capital despite the presence of porous boundaries that favor constant member turnover"? I investigate how open collaborative productions take advantage of a

modular structure to achieve outcomes, such as specialization and knowledge transfer, which are obtained through formal organizational design solutions in more traditional production settings (Simon, 1969; Baldwin & Clark, 2006; Baldwin, 2008). In particular, I observe how organizations are able to retain, and benefit from, participants who tend to specialize their identity within modules and knowledge domains, because other participants find it easier to understand their expertise and their role in the project (Shah, 2006; Fang & Neufeld, 2009). However, participants who reach out and collaborate with the same contributors within or across different modules are able to gain legitimacy without the need to constrain their identity on a narrow knowledge domain. As a result they tend to stay longer in the project. This chapter contributes to the literature on open innovation by identifying the mechanisms that sustain individual participation in open productions, in absence of evident economic incentives.

Taken together, the three studies included in my dissertation help advance our understanding of relational coordination in open collaborative productions by identifying specific processes and mechanisms that facilitate collective problem-solving under extreme conditions of decentralization.

1.5 Empirical setting

Consistent with the literature on F/OSS and open innovation, I focus my attention on bug-fixing and code commits, the two most prominent problem-solving activities in software projects (Crowston and Scozzi, 2008). Bug-fixing refers to the sequence of tasks that are intended to resolve software problems that cause computer programs to behave in unintended and undesirable ways. Code commits refer to submitting the latest changes of the software source code to the repository with the aid of a concurrent version control (CVS) system (i.e., a system devoted to tracking and providing control over concurrent changes to source code).

Using the software Bicho (Robles et al., 2009) and CVSAnalY (Robles et al., 2005), I collected the raw data by parsing the web pages of all relevant bug reports and CVS commits within two F/OSS projects, Epiphany (a web browser) and Apache (a web server application). The raw data were then dumped in a MySQL database and subsequently imported into R - the statistical modelling environment within which I conduct all my analyses. For study 1, I coded the complete set of bug-fixing activities recorded in Epiphany's bug repository during one release cycle of the software (from March to September 2006). Throughout the release cycle 135 developers were active in resolving 719 bugs. All bugs engaged by developers during the release cycle are included and all actions taken while working on them are included. I then coded the complete set of bug-fixing activities (Study 2) and code commits (Study 3) within the Apache HTTPS server between 1996 and 2013. During this period 2630 participants were active in the bug repository attempting to solve 5646 bugs, whereas 111 participants were active in the CVS repository making 1454 commits to modify 10757 software files. Detailed descriptions of the data and the samples are contained in each of the chapters that are structured as independent, self-contained studies.

An important feature of my research design that makes F/OSS a good empirical setting for studying problem-solving in open productions is social transparency (Stuart et al., 2012; Dabbish et al., 2012). Social transparency refers to a new and growing phenomenon affecting collaboration practices over the internet. The contemporary social web provides an unprecedented level of transparency in the form of immediate traceability, and extensive visibility, of participants' history of actions on public or shared artefacts, such as bug or code repositories in F/OSS projects. The decisions of any participant are thus immediately visible to all members of the community and produce information that other participants may take into account in their own process of decision-making. In the case of F/OSS projects, patterns of past interactions between contributors and software bugs or code repositories induce local dependencies that may influence future associations between participants – as bearers of

potential solutions – and problems. This process of endogenous structuration generates the possibility of observing recurrent work practices, or patterns of association between problems and participants, and, indirectly, between participants through problems. Inherent characteristics of participants and problems also influence the likelihood of their association, thus leading to complex patterns that may give rise to core organizational properties such as collaboration, coordination, and specialization (Lomi, Conaldi & Tonellato, 2012).

1.6 Outline of the thesis

The next three chapters are focused on the three empirical studies that I briefly outlined in this introduction. Parts of Chapter 2 are based on the following two papers that I had published with my co-authors:

Conaldi, G., Lomi, A., & Tonellato, M. (2012). Dynamic models of affiliation and the network structure of problem solving in an open source software project. Organizational Research Methods, 15(3), 385-412.

Lomi, A., Conaldi, G., Tonellato, M., & Pallotti, F. (2014). Participation motifs and the emergence of organization in open productions. *Structural Change and Economic Dynamics*, *29*, 40-57.

Chapters 3 and 4 are working papers constituted by unpublished material. In Chapter 5 I draw general theoretical implications based on the three empirical studies, identify current limitations and propose further directions that have the potential to advance our understanding of problem solving in collaborative open productions.

References

Bailey, D. E., Leonardi, P. M., & Barley, S. R. (2012). The lure of the virtual. *Organization Science, 23*(5), 1485-1504.

Baldwin, C. Y. (2008). Where do transactions come from? Modularity, transactions, and the boundaries of firms. *Industrial and corporate change*, *17*(1), 155-195.

Baldwin, C. Y., & Clark, K. B. (2006). The architecture of participation: Does code architecture mitigate free riding in the open source development model?. *Management Science*, *52*(7), 1116-1127.

Baldwin, C., & von Hippel, E. (2011). Modeling a paradigm shift: From producer innovation to user and open collaborative innovation. *Organization Science*, *22*(6), 1399-1417.

Barley, S. R., & Kunda, G. (2001). Bringing work back in. *Organization science*, *12*(1), 76-95.

Benkler, Y. (2006). *The wealth of networks: How social production transforms markets and freedom*. Yale University Press.

Benkler, Y., & Nissenbaum, H. (2006). Commons-based Peer Production and Virtue. *Journal of Political Philosophy*, *14*(4), 394-419.

Bergquist, M., & Ljungberg, J. (2001). The power of gifts: organizing social relationships in open source communities. *Information Systems Journal*, *11*(4), 305-320.

Bonaccorsi, A., & Rossi, C. (2003). Why open source software can succeed. *Research policy*, *32*(7), 1243-1258.

Breiger, R. L. (1974). The duality of persons and groups. *Social forces*, *53*(2), 181-190.

Breiger, R. L. (2000). A tool kit for practice theory. *Poetics*, *27*(2), 91-115.

Breiger, R. L., & Mohr, J. W. (2004). Institutional logics from the aggregation of organizational networks: Operational procedures for the analysis of counted data. *Computational & Mathematical Organization Theory*, *10*(1), 17-43.

Cohen, M. D., March, J. G., & Olsen, J. P. (1972). A garbage can model of organizational choice. *Administrative science quarterly*, 1-25.

Conaldi, G., Lomi, A., & Tonellato, M. (2012). Dynamic models of affiliation and the network structure of problem solving in an open source software project. *Organizational Research Methods*, *15*(3), 385-412.

Crowston, K. (2008). The bug fixing process in proprietary and Free/Libre Open Source Software: a coordination theory analysis. In: Grover, V., Markus, M.L. (Eds.), *Business Process Transformation: Advances in Management Information Systems*. Armonk, NY: M.E. Sharpe.

Crowston, K., Li, Q., Wei, K., Eseryel, U. Y., & Howison, J. (2007). Self-organization of teams for free/libre open source software development. *Information and software technology*, *49*(6), 564-575.

Crowston, K., & Scozzi, B. (2008). Bug fixing practices within free/libre open source software development teams. *Journal of Database Management (JDM)*, *19*(2), 1-30.

Crowston, K., Wei, K., Howison, J., & Wiggins, A. (2012). Free/Libre open-source software development: What we know and what we do not know. *ACM Computing Surveys (CSUR)*, *44*(2), 7.

Cyert, R. M., & March, J. G. (1963). *A behavioral theory of the firm*. Englewood Cliffs, NJ: Prentice-Hall.

Dabbish, L., Stuart, C., Tsay, J., & Herbsleb, J. (2012, February). Social coding in GitHub: transparency and collaboration in an open software repository. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work* (pp. 1277-1286). ACM.

Dahlander, L., & Magnusson, M. (2008). How do firms make use of open source communities? *Long Range Planning*, *41*(6), 629-649.

Dahlander, L., & Gann, D. M. (2010). How open is innovation?. *Research policy*, *39*(6), 699-709.

Dahlander, L., & Wallin, M. W. (2006). A man on the inside: Unlocking communities as complementary assets. *Research Policy*, *35*(8), 1243-1259.

Deci, E. L., & Ryan, R. M. (1987). The support of autonomy and the control of behavior. *Journal of personality and social psychology*, *53*(6), 1024.

Enkel, E., Gassmann, O., & Chesbrough, H. (2009). Open R&D and open innovation: exploring the phenomenon. R&d Management, 39(4), 311-316.

Eysenbach, G., Powell, J., Englesakis, M., Rizo, C., & Stern, A. (2004). Health related virtual communities and electronic support groups: systematic review of the effects of online peer to peer interactions. Bmj, 328(7449), 1166.

Fang, Y., & Neufeld, D. (2009). Understanding sustained participation in open source software projects. *Journal of Management Information Systems*, *25*(4), 9-50.

Feldman, M. S. (2010). Managing the organization of the future. *Public Administration Review*, *70*(1), 159-163.

Ferraro, F., & O'Mahony, S. (2012). Managing the boundary of an "open" project. In J. F. Padgett & W. W. Powell (Eds.), *The Emergence of Organizations and Markets* (545-565). Princeton, NJ: Princeton University Press.

Gavetti, G., Greve, H. R., Levinthal, D. A., & Ocasio, W. (2012). The behavioral theory of the firm: Assessment and prospects. *The academy of management annals*, *6*(1), 1-40.

Ghosh, R. A. (1998). Cooking pot markets: an economic model for the trade in free goods and services on the Internet. *First Monday*, *3*(3).

Gittell, H. J. (2002). Coordinating mechanisms in care provider groups: Relational coordination as a mediator and input uncertainty as a moderator of performance effects. *Management Science*, *48*(11), 1408-1426.

Haefliger, S., Von Krogh, G., & Spaeth, S. (2008). Code reuse in open source software. *Management Science*, *54*(1), 180-193.

Hackman, J. R. & Morris, C. G. (1978) Group tasks, group interaction process, and group performance effectiveness: A review and proposed integration. In L. Berkowitz (Ed.) *Group Processes, volume 8 of Advances in Experimental Social Psychology* (45-99). New York, NY: Academic Press.

Hannan, M. T., & Freeman, J. (1984). Structural inertia and organizational change. *American sociological review*, 149-164.

Hars, A., Ou, S., (2002). Working for free? Motivations for participating in Open-Source projects. *International Journal of Electronic Commerce*, *6*, 25–39

Hertel, G., Niedner, S., and Herrmann, S. 2003. "Motivation of software developers in open source projects: An internet-based survey of contributors to the Linux Kernel," *Research Policy (32*:7), pp. 1159-1177.

Ilgen, D. R., Hollenbeck, J. R., Johnson, M., & Jundt, D. (2005). Teams in organizations: From input-process-output models to IMOI models. *Annual Review of Psychology*, *56*, 517-543.

Jeppesen, L. B., & Frederiksen, L. (2006). Why do users contribute to firm-hosted user communities? The case of computer-controlled music instruments. *Organization science*, *17*(1), 45-63.

Kamel Boulos, M. N., & Wheeler, S. (2007). The emerging Web 2.0 social software: an enabling suite of sociable technologies in health and health care education1. Health Information & Libraries Journal, 24(1), 2-23.

Katz, R. (1982). The effects of group longevity on project communication and performance. *Administrative Science Quarterly*, 81-104.

Lakhani, K. R., & Von Hippel, E. (2003). How open source software works:"free" user-to-user assistance. *Research policy*, *32*(6), 923-943.

Lakhani, K. R., Lifshitz-Assaf, H., & Tushman, M. (2013). Open innovation and organizational boundaries: task decomposition, knowledge distribution and the locus of innovation. *Handbook of economic organization: Integrating economic and organizational theory*, 355-382.

Lanzara, G. F., & Morner, M. (2005). Artifacts rule! How organizing happens in open source software projects. *Actor-network theory and organizing*, 67-90.

Lee, G. K., & Cole, R. E. (2003). From a firm-based to a community-based model of knowledge creation: The case of the Linux kernel development. *Organization science*, *14*(6), 633-649.

Lerner, J., & Tirole, J. (2001). The open source movement: Key research questions. *European Economic Review*, *45*(4), 819-826.

Lerner, J., & Tirole, J. (2002). Some simple economics of open source. *The journal of industrial economics*, *50*(2), 197-234.

Levine, S. S., & Prietula, M. J. (2014). Open Collaboration for Innovation: Principles and Performance. Organization Science. In press.

Levinthal, D. A., & March, J. G. (1993). The myopia of learning. *Strategic management journal*, *14*(S2), 95-112.

Liang, D. W., Moreland, R., & Argote, L. (1995). Group versus individual training and group performance: The mediating role of transactive memory. *Personality and Social Psychology Bulletin*, *21*(4), 384-393.

Lomi, A., Conaldi, G., & Tonellato, M. (2012). Organized anarchies and the network dynamics of decision opportunities in an open source software project. *Research in the Sociology of Organizations*, *36*, 363-397.

Lomi, A., Conaldi, G., Tonellato, M., & Pallotti, F. (2014). Participation motifs and the emergence of organization in open productions. *Structural Change and Economic Dynamics*, *29*, 40-57.

MacIntyre, A. (1981). *After virtue*. London: Duckworth.

MacIntyre, A. (1998). Politics, Philosophy and the Common Good. In K. Knight (Ed.) *The MacIntyre Reader* (235–252), University of Notre Dame Press.

March, J. G., & Simon, H. A. (1958). *Organizations*. Oxford, England: Wiley.

Marengo, L., & Dosi, G. (2005). Division of labor, organizational coordination and market mechanisms in collective problem-solving. *Journal of Economic Behavior & Organization*, *58*(2), 303-326.

Marengo, L., Dosi, G., Legrenzi, P., & Pasquali, C. (2000). The structure of problem-solving knowledge and the structure of organizations. *Industrial and Corporate Change*, *9*(4), 757-788.

McGrath, J. E. (1991). Time, interaction, and performance (TIP): A theory of groups. *Small Group Research 22*, 147-174.

McGrath, J. E., L. Argote. (2001). *Group processes in organizational contexts*. In, M. A. Hogg, R. Scott Tindale, eds. Blackwell Handbook of Social Psychology: Group Processes Blackwell

Malden, MA, 603–627. Michlmayr, M., 2007. Quality improvement in volunteer free and open source software projects: exploring the impact of release management. PhD. dissertation. Centre for Technology Management, Institute for Manufacturing, University of Cambridge (U.K).

Newell, A., & Simon, H. A. (1972). *Human problem solving*. Englewood Cliffs, NJ: Prentice-Hall.

Obstfeld, D. (2005). Social networks, the tertius iungens orientation, and involvement in innovation. *Administrative science quarterly*, *50*(1), 100-130.

Ocasio, W. (2012). Situated attention, loose and tight coupling, and the garbage can model. *Research in the Sociology of Organizations*, *36*, 293-317.

Osterloh, M., and Rota, S. G. 2007. "Open Source Software Development – Just Another Case of Collective Invention?," Research Policy (36:2), pp. 157-171.

Padgett, J. F., Lee, D., & Collier, N. (2003). Economic production as chemistry. Industrial and Corporate Change, 12(4), 843-877.

Padgett, J. F., & Powell, W. W. (2012). The Problem of Emergence. In J. F. Padgett & W. W. Powell (Eds.), *The Emergence of Organizations and Markets* (1-30). Princeton, NJ: Princeton University Press.

Pentland, B. T., Hærem, T., & Hillison, D. (2011). The (N) ever-changing world: stability and change in organizational routines. *Organization Science*, *22*(6), 1369-1383.

Prato, M., & Stark, D. (2013). Peripheral Vision in Financial Markets: How attention networks shape valuation. In *Academy of Management Proceedings* (Vol. 2013, No. 1, p. 15923). Academy of Management.

Raymond, E. (1999). The cathedral and the bazaar. Knowledge, Technology & Policy, 12(3), 23-49.

Reagans, R., Argote, L., & Brooks, D. (2005). Individual experience and experience working together: Predicting learning rates from knowing who knows what and knowing how to work together. *Management science*, *51*(6), 869-881.

Roberts, J. A., Hann, I., and Slaughter, S. A. 2006. "Understanding the Motivations, Participation, and Performance of Open Source Software Developers: A Longitudinal Study of the Apache Projects," Management Science (52:7), pp. 984-999.

Robles, G., Gonzalez-Barahona, J. M., Izquierdo-Cortazar, D., & Herraiz, I. (2009). Tools for the study of the usual data sources found in Libre Software projects. International Journal of Open Source Software & Processes, 1(1), 24-45.

Scacchi, W. (2004). Free and open source development practices in the game community. *Software, IEEE*, *21*(1), 59-66.

Shah, S. K. (2006). Motivation, governance, and the viability of hybrid forms in open source software development. *Management Science*, *52*(7), 1000-1014.

Simon, H. A. (1947). *Administrative behavior.* New York, NY: Macmillan.

Simon, H. A. (1965). The architecture of complexity. *General systems*, *10*(1965), 63-76.

Simon, H. A. (1969). *The sciences of the artificial*. Cambridge, MA: MIT press.

Stewart, D. (2005). Social status in an open-source community. American Sociological Review, 70(5), 823-842.

Stewart, K. J., and Gosain, S. (2006). The Impact of Ideology on Effectiveness in Open Source Software Development Teams. *MIS Quarterly*, 30(2), 291-314.

Stuart, H. C., Dabbish, L., Kiesler, S., Kinnaird, P., & Kang, R. (2012). Social transparency in networked information exchange: a theoretical framework. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work* (pp. 451-460). ACM.

Sullivan, B. N. (2010). Competition and beyond: Problems and attention allocation in the organizational rulemaking process. *Organization Science*, *21*(2), 432-450.

van Der Vegt, G. S., & Bunderson, J. S. (2005). Learning and performance in multidisciplinary teams: The importance of collective team identification. *Academy of Management Journal*, *48*(3), 532-547.

von Hippel, E., & von Krogh, G. (2003). Open source software and the "private-collective" innovation model: Issues for organization science. *Organization science*, *14*(2), 209-223.

von Krogh, G., Spaeth, S., & Lakhani, K. R. (2003). Community, joining, and specialization in open source software innovation: a case study. *Research Policy*, *32*(7), 1217-1241.

von Krogh, G., & von Hippel, E. (2006). The promise of research on open source software. *Management science*, *52*(7), 975-983.

von Krogh, G., Haefliger, S., Spaeth, S., & Wallin, M. W. (2012). Carrots and rainbows: Motivation and social practice in open source software development. *Mis Quarterly*, *36*(2), 649-676.

West, J. (2003). How open is open enough?: Melding proprietary and open source platform strategies. *Research policy*, *32*(7), 1259-1285.

West, J., & Gallagher, S. (2006). Challenges of open innovation: the paradox of firm investment in open-source software. R&D Management, 36(3), 319-331.

Zeitlyn, D. (2003). Gift economies in the development of open source software: anthropological reflections. *Research policy*, *32*(7), 1287-1291.

Chapter 2

RELATIONAL COORDINATION IN AN OPEN SOURCE SOFTWARE PROJECT: THE ROLE OF ATTENTION NETWORKS

ABSTRACT

Drawing from insights of a structural perspective within the attention-based view of the firm and social cognition, this paper argues that the process through which organizational members self-assign to competing problems in open productions unfolds through the interdependent acts of attention allocation linking members and problems. Hence attention processes are not individual, but embedded in the evolving portfolio of problems to which a member is attentive and in the evolving attention structures of other members to whom he or she is tied. In this way multiple members are linked to multiple problems by attention networks. I analyze the bipartite dynamic structure of these networks by proposing and investigating two fundamental mechanisms that sustain a situated view of attention: attention clustering and attention spread. I explore these mechanisms by examining problem-solving attempts performed by 135 participants in an open source software project on the 719 problems (software bugs) recorded during a complete release cycle of the software. I find that attention allocation decisions show a tendency towards clustering of related problems and towards disassortative attention spread, and that these two effects are driven by the high levels of experience of problem solvers. The discussion focuses on the broad theoretical implications of the study for the relationship between attention and problem-solving in open productions.

2.1 Introduction and motivation

When organizational participants are involved in the process of finding solutions to problems, attention becomes crucial as decision-makers tend to concentrate organizational resources to areas where attention has been allocated (March & Simon, 1958; Sullivan, 2010). However, the task of spreading attention over a broad spectrum of organizational problems is complex. Instead of trying to uniformly distribute the limited attention across all problems, as if they were equally important to the organization (Bouchet & Birkinshaw, 2008), effective problem-solving practices require that organizational participants are selective in their efforts. As attention has always being recognized as a scarce resource in organizations (Simon, 1947; Ocasio, 2011), the principle of selective attention allocation across competing problems becomes crucial for achieving effective problem-solving. But how is this process structured?

Extant research focuses on the role that formal structures play in guiding members' attention toward the most salient problems and the most attractive opportunities (Ocasio, 1997; Ocasio & Joseph, 2005). This *top-down* approach – driven by goals, schemas, and rules – links attention to cognitive primers. According to this view participants are not perfectly rational individuals, but individuals who are embedded in a social and cultural system, based on cognitive schemas that are often taken for granted and not necessarily optimal from a purely economic standpoint. Actors are anchored to categorical interpretations of the task environment, each rooted in established institutional logics that drive attention and ultimately action (Fligstein, 2002; Lounsbury, 2007; Thornton & Ocasio, 2008). A second, competing stream of research investigates how certain characteristics of problems render them visible or accessible beyond the constraints of these top-down procedural structures, to capture the attention of individual members more effectively (Hansen & Haas, 2001; Sullivan, 2010). This *bottom-up* approach is stimulus-driven and links attention to characteristics of the stimuli that are

to be attended to (Corbetta & Shulman, 2002). An example of the bottom-up mechanism is represented by the effect of problem salience, or "urgency" (Sullivan, 2010) on the likelihood that the problem is engaged with by organizational decision-makers. Although the two separate processes have been extensively tackled in the literature on organizational attention, the ecological mechanisms underlying attention competition haven't received enough theoretical and empirical investigation (Sullivan, 2010). In particular, while scholars belonging to the Carnegie School tradition have generally argued that attention competition is situated in a particular context, there is a research gap regarding the specific attention allocation mechanisms operating within this contextual structure articulated as an ecology of people, problems and solutions (Cohen, March & Olsen, 1972).

If top-down and bottom-up exogenous, stand-alone mechanisms of situated attention have been the focus of a recently burgeoning literature on organizational attention (for a review see Ocasio, 2011), I argue that, with a few notable exceptions, the idea of *attention networks* of interdependent acts of attention allocation has been largely overlooked in recent empirical work. Drawing originally on March and Olsen's (1976) intuition that one's attention is a function of other's attention, this view posits that the attention allocation patterns of one person are *contingent* on, and *situated* in, a structure of interdependent actors, problems and solutions. More recent work rooted in a situated cognition perspective has further advanced this intuition. Most notably, Prato & Stark (2013) showed that financial analysts' estimates of stocks' earnings per share are shaped by the other stocks that populate their field of view. Furthermore, their analysis revealed that analysts' valuations are also influenced by the attention spectrum of those neighboring analysts that are connected through coverage of shared stocks. These findings suggest that individual estimates of a stock don't happen in a vacuum but are embedded in the broader attention structure that connects analysts and stocks. In other words, individual valuations of a focal stock are influenced by the valuations of "neighboring" individuals, not only on the focal stock, but also on peripheral stocks.

Against this backdrop this paper presents an alternative approach to the main body of existing literature on organizational attention, using the emerging evidence of the ecologically situated nature of attention mechanisms as its point of departure. This approach focuses on how individual decisions about whether to engage with an organizational problem or not actually happen as a consequence of dynamic attentional processes linking individuals and problems in organizations (Conaldi, Lomi & Tonellato, 2012). More specifically, I examine how problems and individuals encounter each other in a space of potential attention opportunities and how such encounters are both affected by, and give rise to, "attention networks", or dependence structures of attentional engagement decisions in which individuals and problems are embedded. According to this view, the attention of organizational members is embedded in the organization's network of tightly and loosely coupled decision-making channels defined as "the formal and informal concrete activities, interactions, and communications set up by the firm to induce organizational decision-makers to action on a selected set of issues" (Ocasio, 1997, p. 194). In the process of evaluating which organizational problem to engage with, individuals use these channels as a source of information about other individuals' previous attention allocation choices. I argue that the bipartite network linking individuals to problems internalizes the information produced by individual acts of problem-solving and by constantly letting other members access this information to decide which problems to select next. The global structure of this situated attention system emerges from individual decisions of problem selection, which consolidate into regular patterns of network ties. I present two attentional mechanisms that match these regular configurations of ties to meaningful properties pertaining to the structural distribution of attention in open productions: attention clustering and attention spread.

These two distinctive structural features of the attention network deserve closer scrutiny as they reveal underlying forms of micro-behavioral mechanisms. The first mechanism I identify is *attention clustering* of related problems – the tendency of organizational members to concentrate their attention

on groups of problems linked to the same set of individuals. Organizational members tend to select problems embedded in attention clusters because they can make better evaluations of their saliency due to easier comparison and categorization processes (Smith & Collins, 2009; Zuckerman, 2004). Furthermore, by selecting problems embedded in attention clusters, organizational members can better build on prior learning developed about the same, familiar individuals (Tsai, Su & Chen, 2011). In the network linking organizational participants to organizational problems the tendency toward attention clustering is captured by the recurrence of patterns of bipartite closure (Wang et al., 2012).

The second feature is *attention spread*, which captures the tendency of organizational problems that have already attracted attention of many active members to be progressively more (or less) likely to attract additional active members (Barabási & Albert, 1999). A disassortative *attention spread* (a tendency against assortativity) would be evidence that participants prefer to allocate their attention to problems that are not already attended to by other participants. The possible outcome of a disassortative attention spread would be a more balanced allocation of attention over available problems. In the context of our argument disassortativity is important because in hierarchical organizations this outcome is frequently obtained by formal mechanisms of attention allocation based, for example, on hierarchical problem-scheduling routines.

I then discuss how traditional top-down mechanisms governing attention allocation decisions, such as reinforcing cognitive schemas and reputation-based competition induced by organizational members' experience, moderate the above-mentioned network configurations. In particular, I show how reinforcing schemas associated with experience tend to induce tenured individuals to allocate their attention to increasingly familiar problems (i.e., embedded in clusters of related problems) while reputation concerns tend to induce experienced individuals to engage with less popular problems (i.e., problems that show a tendency toward a disassortative attention spread).

My contributions to the organizational literature are twofold. Firstly, I contribute to the burgeoning debate on situated cognition by extending the concept of attention networks through the identification of different mechanisms showing the situated nature of attention allocation processes. According to this view, problem solvers' selection of tasks at hand are intertwined with, and affected by, attention to decisions of fellow individuals to which they are indirectly tied through shared problems. Secondly, I contribute to the literature on organizational networks by providing new lenses through which to investigate the dynamics of bipartite attention ties on individual behavior and cognition. Attention networks are bipartite (two-mode) structures linking individuals to problems that they attend to. The vast majority of the literature on organizational networks reduce two-mode structures to one-mode networks. For example, the two-mode association between companies issuing securities and the investment banks managing the issues (Podolny, 1994) may be decomposed into two one-mode associations. The first is between companies affiliated indirectly through the banks managing their issues. My modeling efforts avoid this artificial reduction by directly modeling indirect ties between actors and their effect on behavior (Conaldi et al., 2012).

2.2 Illustration of research issues in the empirical context

Consistent with the notion of situated attention (Ocasio, 1997) and distributed social cognition (Smith & Collins, 2009), in this paper I explore the mechanisms through which individuals self-assign to problems in open productions. In particular I investigate the effect of attention networks on individual decisions to select problems to solve within a large Free/Open Source Software (F/OSS) project named "Epiphany". F/OSS projects are organizations that exploit the (geographically) distributed knowledge of a team of (mainly volunteer) contributors in order to produce new software. The setting is particularly apt because it exemplifies a context in which an organization faces problems

to which seeks and generates solutions when exogenously established hierarchical structures and formal coordination mechanisms are weak. Given the weakness of centralized control, and the direct access of participants to problems, all the basic components of organizational attention structure are endogenous and regulated by processes of self-selection and self-assignment. More specifically, the model is applied to the dual association between software problems (or software "bugs") and software contributors observed throughout one complete release cycle of Epiphany (Conaldi & Lomi, 2013). Figure 2.1 visualizes my idea of an attention network by displaying the actual data that I analyze in the empirical part of the paper. Blue squares represent project contributors and red circles represent software bugs.

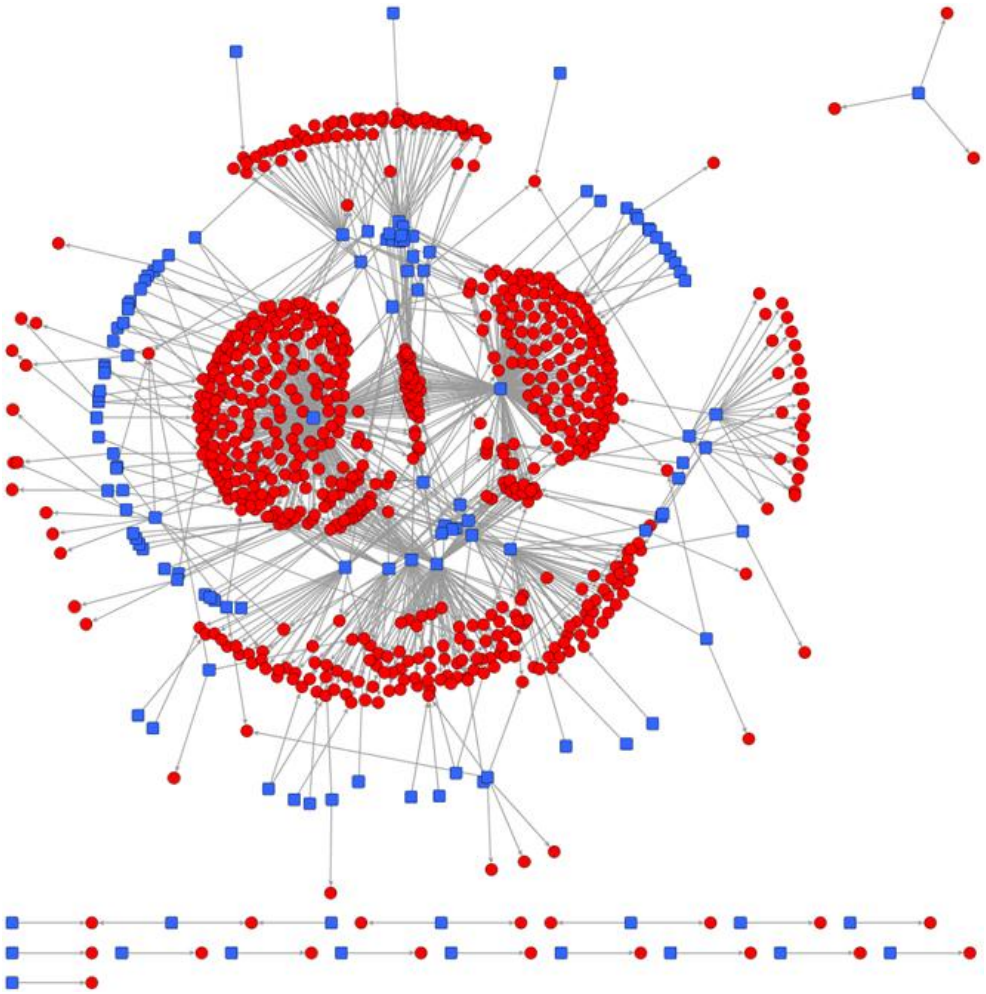


Figure 2.1: The observed organizational attention network. Blue squares are members. Red circles are problems.

Figure 1 suggests a core-periphery pattern with a small number of core participants engaged in the resolution of a large number of problems, and a larger periphery of participants characterized by a much lower level of engagement in organizational problem-solving activities (Crowston et al., 2006)

According to this network representation of organizational problem-solving, organizational problems and organizational participants are linked by a dual relationship (Breiger, 1974): on the one hand problems (software bugs in our case) are connected through the set of organizational participants (software developers in our case) jointly working on their resolution; on the other hand organizational participants are connected through the problems they jointly engage with. This idea of duality is central to the study of organizations where the identity of units at one level is frequently defined in terms of patterns of association at a different – higher or lower – level (Breiger, 2000). The association between individuals and tasks, and between individuals and knowledge, for example, are typically considered the fundamental building blocks of organizations (Carley, 1991). These associations are "dual" in the sense that they connect (and indeed "define") social entities standing in a mutually constitutive relation (Breiger & Mohr, 2004). In empirical research the mutual constitution of social entities across levels of organizational analysis frequently takes the form of a bipartite network (Pattison & Breiger, 2002).

Bug-fixing has long been recognized as playing an essential role in the development and maintenance of F/OSS products (Crowston, 2008). Decentralized bug-fixing activities are frequently viewed as a core organizational problem-solving activity providing a significant contribution to the success of F/OSS projects (Crowston & Scozzi, 2008). Processes of resolution of software bugs in F/OSS projects have attracted considerable attention recently and represent a well-recognized area of interest in the broader context of research on the social organization of open productions (Zanetti et al., 2013). While bug-fixing is obviously only one of the many problem-solving activities carried out

within F/OSS projects, it is representative of the decentralized, user-centered process of quality control and improvement that is typically observed in F/OSS productions (Conaldi & Lomi, 2013). From this perspective bug-fixing represents an essential element in the overall quality control process, ensuring quality improvement between successive releases (Michlmayr, 2007).

2.3 Theory and hypotheses

This study arises in response to the fundamental question of how organizational participants allocate attention resources to solve problems. Research on decision-making and problem-solving in organizations has often taken the position that problems trigger the search for solutions (March & Simon, 1958; Cyert & March, 1963; Thompson, 1967). However, in a world of bounded rationality, organizational problem-solvers don't keep searching until the evaluated optimal solution is found; rather they pick the first satisfactory choice. In other words, individuals look for solutions to problems locally, in the "neighborhood" of the problem under attention. (Cyert & March, 1963; Winter et al., 2007). When individuals are involved in problem-solving activities, organizational attention becomes the most crucial factor for decision-making processes, as problem solvers devote increasing levels of cognitive and physical resources to regions of problems that have attracted most attention. Scholars who belong to the so-called Carnegie School have thoroughly demonstrated that, when looking for solutions to problems at hand, organizational participants do not engage problems uniformly, because they have limited attention capacity (Simon, 1947; March & Simon 1958). This situation leads to a scenario in which boundedly rational individuals and organizations have to allocate their limited attention to problems sequentially (Cyert & March, 1963). As in any decision on resource-allocation, there is a trade-off: the gain in attention by one problem domain means a concomitant loss of attention by another, and organizational participants are more likely to generate solutions in problem areas that

have already are attracted a substantial amount of attention. As a matter of fact, multiple, neighboring problems compete for participants attention.

Building on the Carnegie School legacy, Ocasio (1997) advanced a perspective called the attention-based view (ABV) of the firm. The central argument of the ABV (Ocasio, 1997, p. 188) is that "to explain firm behavior is to explain how firms distribute and regulate the attention of their decision-makers". Broadly speaking, the ABV describes and explains how organizational responses are shaped by the manner in which organizations devote attention to their environments and by the way in which stimuli are distributed and channeled into decision-making processes (Ocasio, 1995). Organizations generate attention-directing cues that guide the behavior of individuals in a context of bounded rationality. Besides, to help channeling participants' attention towards certain areas of the organizational environment and not others, firms rely on certain organizational design features, such as hierarchical systems, functions, departments, communication channels. The seminal paper that presents the ABV, proposes a set of constructs and connecting mechanisms that "explicitly link[s] individual information processing and behavior to the organizational structure through the concepts of procedural and communication channels and attention structures" (Ocasio, 1997, p. 188). In particular the concepts of *situated attention* and *structural distribution of attention* prove crucial in the context of this study. According to the former principle, organizational participants' focus of attention depends on the features of the situation in which they are embedded. According to the latter, the specific situation in which participants are embedded, and how they attend to it, is affected by the way in which the organization regulates and allocates problems, solutions, and problem-solvers to particular functions, communication channels, and routines. In other words, organizations create structures that affect what problems attract participants' attention, the solutions available to them to solve these problems, and eventually the decisions they make. Together, these two principles constitute a *structural view of attention* which helps explain how the problems that attract participants' attention depend on the way in

which organizations design specific structures.

Traditionally, research on ABV has demonstrated that organizations contextual and procedural attention structures influence the business opportunities identified by decision-makers by affecting the way in which they allocate their limited attention on proximate, relevant stimuli (Barnett, 2008). At the intra-organizational level, Williams & Mitchell (2004) found that business opportunities such as new market entry pursued by managers are enabled by the infrastructure of organizational information, conceptualized as the way in which organizations design the communication links between their subunits. Furthermore Cho & Hambrick (2006) found that the attention of managers employed by airline companies moves from an internal to an external market orientation as a result of a shift from a regulated to a deregulated industry, exemplifying what Ocasio (1997) calls a "change in the rules of the game", that opens the leeway to exploration of new ventures. Similarly, Yu et al. (2005) explored the effect of different contextual structures on the likelihood to shift managerial attention away from the current set of issues. The authors in particular explored the role of turnover, leading to the introduction of new members into the decision structure, and how this inception of new attention schemas lead the company to focus its attention on a more novel set of problems associated to their merger integration process.

However, the emergence of open, collaborative forms of economic production (Bailey et al., 2012; Levine & Prietula, 2014), such as open source software projects, invites reconsideration of the relationship between organizational structures and actual attentional patterns, giving rise to what has been labeled a theory of "social transparency" (Dabbish et. al., 2012). In open production, most work practices are carried out using social applications that let members track and follow the activities of other members, irrespective of their location. In open source software this new approach mixes version control systems with features of social media to create transparent work environments, where every

action undertaken by any individual is immediately visible and traceable by other project participants (Dabbish et al., 2013). Social transparency indicates the fact that origin and history of all actions are visible to other participants. Contributors to open productions keep everyone up to date on things they do or work on and in turn decide which individuals or problems of interest they want to pay attention to. Actions from tracked individuals and problems appear in each contributor's feed, thus shaping the overall attention network.. In our example Bugzilla stores the complete history of every change that has been applied to each bug report. Since social transparency allows participant to be aware of what feature of the project is being modified, when, where and by whom, this meta-information is used by other participants to coordinate their efforts and respond to changes in content appropriately (Stuart et al., 2012). I argue that these type of transparent cues will generate direct and indirect effects on the attention allocation mechanisms of other members, and their consequent decisions about problems self-assignment. For instance, software contributors can now infer someone else's technical expertise when they co-edit a bug report, or guess which of several similar problems has the best chance of being solved in the short term. Contributors combine these attentional inferences into effective strategies for coordinating work and selecting tasks.

Recent advancements in social psychology and theories of managerial cognition highlight the relevance of the situated context in which cognitive processes are embedded (Smith & Collins, 2009). A further development along these coordinates introduced the concept of situated attention, according to which individual acts of attention allocation are situated in, and influenced by, an ecology of individuals and situations to which they are compared (Ocasio, 2012). If an organizational member's situated attention is ultimately a function of other members' attention, I can argue that these multiple dependencies give rise to attention networks, where transparent attentional cues are used by individuals to assess and decide their own level of engagement in problem-solving activities. I conceptualize and derive hypotheses on these endogenous attentional mechanisms in the following section.

2.3.1 Attention clustering

When considering problem-solving activities in open productions the most striking organizational feature is the emergence of sustained collaboration around related problems. Adopting a situated attention approach, I argue that a problem-solver's decisions are not merely the outcome of matching characteristics of the focal problem with one's goals and expertise; they are always evaluated against the backdrop of other concurrent acts of attention allocation. That is, the multiple problems across which multiple problem-solvers allocate their attention will profile the characteristics that are considered as salient and meaningful when evaluating whether to attend to the focal problem or not.

According to Goldberg (2011: p. 2), the "meanings that social actors attribute to symbols and actions emerge from the multiple associations they make between them". This suggests that the evaluations individuals make of the problems they are facing should be understood as having a relational dimension (Mohr, 1998; Emirbayer, 1997). When assessing individuals' attention allocation decisions we should not only take into account the decisions they take on specific problems independently, but rather the relationships between those decisions and a series of other elements that make up the specific socio-cognitive domain in which they are embedded. Hence, when organizational members evaluate the saliency of a problem they consider the number and characteristics of other problems to which it is related (Piezunka & Dahlander, 2014).

A problem is more strongly related to another problem the more individuals exclusively engage with both of them, suggesting – for instance – that the two problems are evaluated on a common metric, possess common features, or are linked by relationships of complementarity. This type of clustering of problems channels organizational member's attention because it helps to facilitate their understanding of the focal problem. Thus, if a problem is related to other problems – through shared acts of attention allocation – organizational members can better assess its salience and characteristics,

by evaluating them on the backdrop of other problems that are cognitively close. For instance, this process might enable the creation of heuristics that ease information-processing; that is, the establishment of labels and categories that facilitate a shared understanding of the problem under focus (Piezunka & Dahlander, 2014). The presence of related problems increases the likelihood that an individual will engage with a focal problem because organizational members are better equipped to make a sound assessment of its salience and characteristics.

It is well known and has been empirically demonstrated that organizations tend to search for solutions locally (Cyert & March, 1963; Levinthal & March, 1993; Stuart & Podolny, 1996). However, we know less about how organizations sequentially and proactively look for problems to solve. When attending to one problem, individuals are likely to later shift their attention to a related problem where they can build on experience of prior collaborations with the same individuals. In bug-fixing, where writing "patches" to bugs is considered the most time-consuming action, code developers tend to improve time efficiency by engaging with bugs that were triaged by trusted individuals, whose efforts were likely to be recognized as valuable in past collaborations. This idea of local search suggests that individuals' problem engagement choices take advantage of the proximity of new problems clustered with other problems through common individuals with whom they are already affiliated.

I characterize attention clustering using the network structure that links problems and problem solvers. In a bipartite affiliation network, direct ties between objects of the same kind are not allowed. Ties are only possible between objects belonging to different sets (Easley & Kleinberg, 2010). As a consequence, in bipartite affiliation networks cycles of odd lengths cannot occur: the shortest possible cycle in an affiliation network is a four-cycle. Clustering in a bipartite affiliation network, therefore, may only be defined in terms of a relation linking pairs of different kinds of objects – in my case participant-problem dyads. For this reason I consider the four-cycle as the elementary component of

clustering of related problems. Figure 2.2 illustrates a four-cycle configuration. At time *t*, individuals *i* and *j* attend to problem p_1. Individual *i* attends alone to problem p_2. At time *t+1 i* and *j* extend their attention also to problem p_2. The result is a four-cycle structure which reveals the tendency toward the clustering of attention across shared problems.

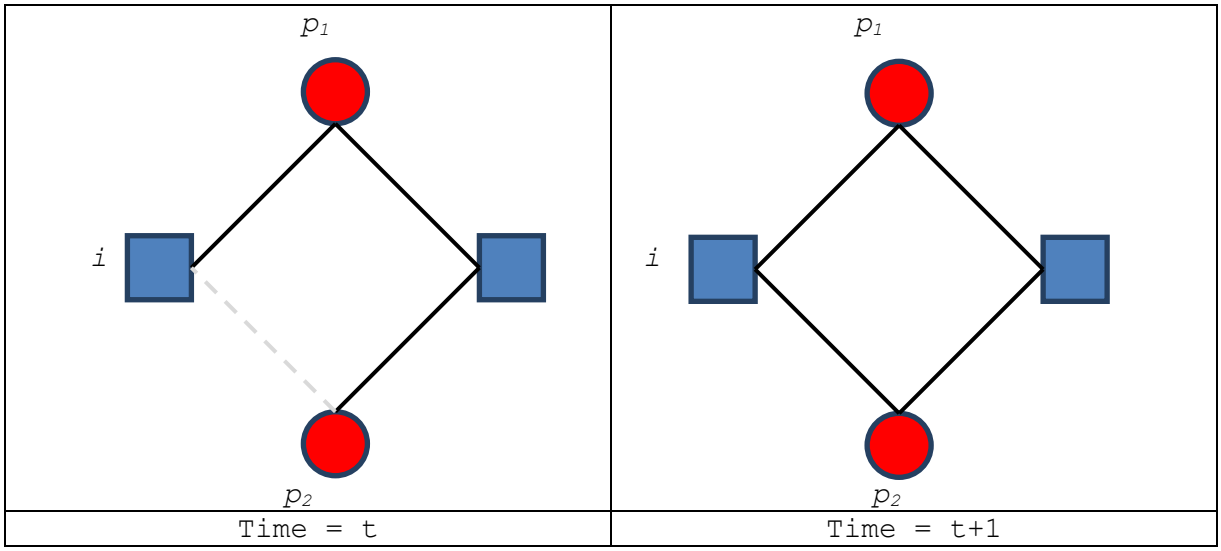


Figure 2.2: Attention Clustering

Like the notion of neighborhood in social networks, the four-cycle configuration in bipartite affiliation networks is important for our current purposes because it defines a "setting" – or a *cluster* – for situated attention towards organizational problems that are linked indirectly through the individuals attending to them (Pattison & Robins, 2002). The presence of four-cycles in a problems-individuals attention network reveals self-organizing tendencies toward sustained collaboration.

Hypothesis 1 (H1): The more a problem is clustered with other problems, linked by attention ties to the same set of individuals, the higher the likelihood that problem solvers will attend to that problem.

2.3.2 Attention spread

While the attentional process of clustering of a problem with other problems increases the chances of an encounter with, and a more accurate evaluation of, a problem, another fundamental mechanism shapes the way in which attention networks influence individuals' problem selection decisions. In organizational members' evaluations of whether or not to select a problem to solve, it also matters how many other organizational members a problem is affiliated to and, dually, how many problems the focal member has already engaged with. I can thus distinguish problems according to whether they attract attention from focused or active members. I can also classify individuals according to whether they prefer to spread their attention to already popular problems – those that have attracted the attention of a large, diversified set of individuals – or to problems which are of a more confined, specialist interest. I refer to a network configuration as showing a tendency toward *assortative attention spread* if a problem attracts the attention of a multiple set of individuals who in turn are attentive to many different problems. Figure.2.3 shows assortative attention spread in detail.

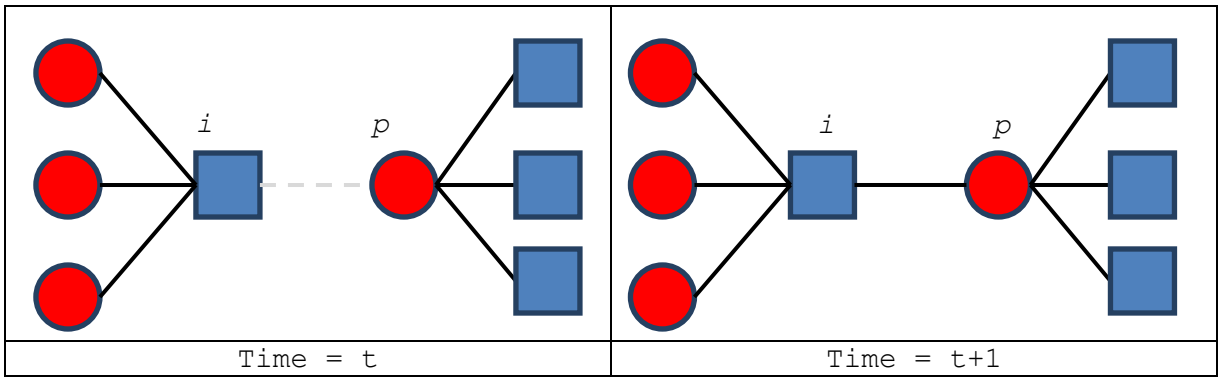


Figure 2.3: Attention Spread

At time *t* participant *i* is very active, or has high out-degree (because he or she spreads her attention to many problems) and problem *p* is very popular, or has high in-degree (because it attracts the attention of many participants). At time *t* + 1 participant *i* decides to engage with problem *p* thus

determining a situation whereby attention is preferentially allocated to more popular problems – a form of preferential attachment (Barabási & Albert, 1999).

Problems that attract a *disassortative attention spread* reach into more distant knowledge domains of the organization by appealing to individuals who spread their attention on a multitude of different types of problems, regardless of their popularity among problem solvers. The degree of assortativity of attention spread affects how a focal organizational member evaluates a problem, which influences the likelihood that he or she decides to attend to it. Attracting a large number of individuals who in turn spread their attention on a large number of problems might hinder information-processing. A problem that attracts contributions from a large number of individuals can potentially benefit organizational performance as it provides access to a more diverse spectrum of expertise that these individuals bring to the problem-solving process. At the same time, however, the more heterogeneous the contributions to the focal problem are, the more difficult it is to categorize it (Zuckerman, 1999; Hsu, 2006). If a problem cannot be clearly evaluated, it is more challenging for the organization to coordinate work around its resolution. This also means that tasks and division of labor within the community are likely to be unclear with respect to the problem, making it more difficult for the organization to assign the problem to an appropriate person for evaluation. A disassortative attention spread (a tendency against direct connection between participants with high outdegree and problems with high indegree) would be evidence that participants prefer to allocate their attention to problems that are not already attended to by many other participants. The possible outcome of disassortative attention spread would be a more balanced allocation of attention over available problems. In bug-fixing, for instance, a bug that is re-assigned multiple times attracts more heterogeneous contributions from a more diverse set of individuals, making it less likely to get eventually fixed (Guo et al., 2010). The most likely end result is that trustworthiness of information drops and the bug is left increasingly unattended to.

Hypothesis 2 (H2): The greater the degree of assortative attention spread a problem attracts, the lower the likelihood that problem solvers will attend to it.

2.3.3 Interaction hypotheses: the role of individual experience

I posit that endogenous attention mechanisms of problem selection are influenced by top-down cognitive schemas that derive from organizational experience. For instance, excessive experience may lead organizational members to overly focus on existing solutions and habitual routines (Cohen & Bacdayan, 1994), which in turn discourages a more wide-ranging search for novel solutions in an ever-changing landscape of problems. Thus, competency traps (Levitt & March, 1988) sometimes make group members reinforce their schemas and stick to familiar socio-cognitive environments where they minimize the level of cognitive dissonance (Festinger, 1962). Furthermore, shared mental models developed in earlier repeated collaborations may induce organizational participants to withhold ideas that could alter the status quo and help maintaining conditions under which the exploration of more novel and riskier solutions is discouraged (Klimoski & Mohammed, 1994; Stasser & Titus, 1985; Cohen & Levinthal, 1990). In the framework of this study this process generates a mechanism by which experienced individuals tend to overly rely on attention clusters when they decide which problem to engage with. Sticking to clusters of problems attended to by the same individuals induces organizational members to tackle problems that are less ambiguous and diverge less from what they have already learned by collaborating repeatedly with the same subset of people.

In the bug-fixing context where a great deal of distributed expertise coordination is required, experienced developers tend to develop shared mental models (SMM) of problem-solving routines (Klimoski & Mohammed, 1994) and transactive memory systems (TMS) of "who knows what" in the team (Liang, Moreland & Argote, 1995) as the result of previous transparent interactions and shared work experience. SMM and TMS facilitate coordination and problem-solving by developing a shared

agreement of where specific expertise is located in the community. Hence, as SMM and TMS consolidate over time due to increased experience of working together, I expect experienced contributors to attend to clusters of related bugs more than inexperienced contributors.

Hypothesis 3a (H3a): Individual experience moderates the relationship between attention clustering of problems and problem selection, such that the higher is an organizational member's experience, the more likely that member is to allocate her or his attention to clusters of related problems.

In the context of open productions, contributors with high experience tend to advance towards the center of the projects, embracing roles with higher coordination duties (Dahlander & O'Mahony, 2011). Those in coordinating roles are expected to be very active and to have a wide attention span in order to tap into the highest possible proportion of reported problems. However, they are less likely to engage with bugs that have already attracted a great deal of attention, leaving them to the discussion of specialized members with more idiosyncratic expertise in that specific problem. Besides, they carry the necessary authority to initiate discussion around bugs that are otherwise left unattended.

Hypothesis 3b (H3b): Individual experience moderates the relationship between attention spread and problem selection, such that the higher is an active organizational member's experience, the less likely he or she is to attend to popular problems.

2.4 Research design

2.4.1 Data

I collected data on the complete set of problem-solving actions recorded during one release

cycle of *Epiphany,* a successful F/OSS project started in 2002 with the goal of building a web browser for the GNOME graphical desktop environment. I collected information on every problem-solving attempt observed from March to September 2006 by parsing the web pages of all relevant bug reports in Epiphany's official bug repository – used by contributors to track all problems affecting the software during development – with the specialized software, *Bicho* (Robles et al., 2009). A more detailed description of the empirical setting and data collection methods is presented in Conaldi et al. (2012).

During the release cycle I recorded a total of 135 contributors allocating their attention over 719 software bugs. I constructed the participant-by-problem matrix by coding every action undertaken by contributor *i* on software bug *p* as a tie linking *i* to *p*. I subdivided the observation period into four time panels ($t_{1\text{-}4}$), each covering a period of approximately 45 days, and recorded all the changes in the network of ties linking individual contributors and software bugs. For example, if contributor *i* engages with software bug *p* only during the first month of the cycle, the tie linking *i* to *p* will be present in the network at t_1 but will be absent from the network in the following panels. I used the Jaccard coefficient to measure the amount of pair-wise stability in the panels (Snijders et al., 2010). The three Jaccard coefficients relative to the four panels in our dataset vary between 0.29 and 0.49. This means that between 29% and 49% of the ties linking contributors and software bugs remain unchanged in the transitions between successive time panels, while the remaining ties change.

2.4.2 Variables and measures

The dependent variable in this study refers to decisions of contributors to self-assign to specific problems to solve. Therefore I model problem selection as the likelihood of contributor *i* to engage bug *j* at a particular time *t*. Acts of problem solving give rise to attention networks because once a bug is first engaged by a contributor all subsequent activities undertaken on the same bug will be fed to the

focal contributor. Therefore, I can posit that when contributor *i* engages with bug j he or she is going to pay attention to all future activities on *j*. I represent attention networks as participant-by-problem matrices by coding every problem-solving act undertaken by contributor i on software bug j as a tie linking i to j. My empirical model specification is based on an attempt to characterize problem selection patterns in terms of (i) baseline change rates; (ii) attentional mechanisms of theoretical interest; (iii) other endogenous network effects; (iv) exogenous individual covariates, and (v) interactions between attentional mechanisms and individual covariates.

Change rates control how quickly new opportunities arise for changing network ties linking individuals to problems. *Baseline change rates* refer to the average number of opportunities that individuals face to change their affiliation to problems, either by engaging with a new problem or by abandoning prior problems. As I discuss below, I include the effect of *Core developer on rate* to control for the additional choice opportunities that core developers enjoy as contributors "who contribute most of the code and oversee the design and evolution of the project" (Crowston & Howison, 2005: 7).

Attentional mechanisms of theoretical interest are associated with self-organizing properties which represent the main focus of the study. They capture systematic dependencies created by the dual association of participants and problems. A summary of these mechanisms can be found in Table 1.

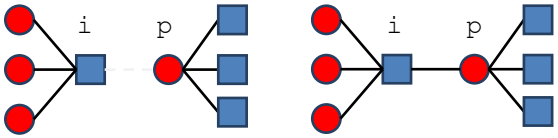
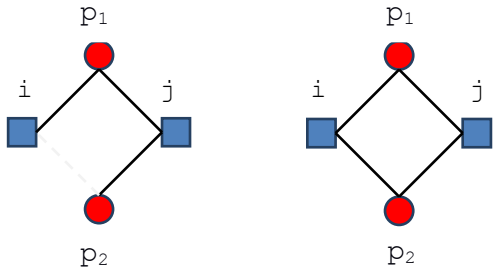
I include three endogenous network effects in addition to those represented by the two mechanisms that I have discussed (clustering of problems and assortative attention spread). The first is *Outdegree (density)* which I include to capture the baseline tendency of contributors to allocate their attention to software bugs, i.e., the likelihood that contributor *i* will engage with software bug *p* at all. I expect the estimate of the associated parameter to be negative to reflect the fact that problem-solving activities are costly. The second is *Software bug popularity* which captures the tendency of software

bugs that are already engaged with by many contributors to be progressively more (or less) likely to attract additional contributors. The third is *Contributor activity*, which I include to capture positive feedback dynamics associated to "learning by doing" (Argote & Epple, 1990). A positive coefficient would indicate that contributors who engage with many sofware bugs are progressively more likely to engage with additional bugs. These additional forms of local dependence are summarized in Table 2.

Exogenous covariates may refer to characteristics of both individuals (e.g., experience), as well as problems (e.g., severity). Exogenous individual (or "participant-specific") covariates are included to control for the tendency of differences among problems and among participants to affect individual problem-solving attempts. Among exogenous actor-specific covariates referring to software contributors, *Core developer* is included to control for the well-known tendency of core developers to perform higher levels of activity within open source projects. *Contributor experience* is incorporated to control for the effect of tenure in the project, whereas *Contributor CC'ing only* is included to tease out the effect of being active in the project only by being put in Carbon Copy (CC) of a bug report.

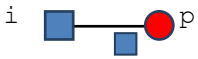
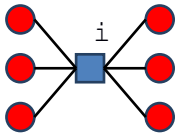
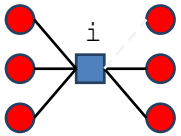
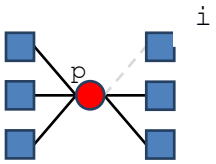
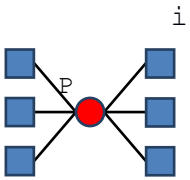
Among exogenous problem-specific covariates, *Software bug severity* is included to control for the propensity of contributors to engage with software bugs that are considered more urgent or important by contributors. Within the project *Software bug severity* is classified on a seven-point scale ranging from "enhancement" (low) to "blocker" (high). I considered a software bug as *severe* if it was assigned to the two highest levels in the scale. *Software bug communication* refers to the number of comments attached to the bug report, whilst *Software bug CC'ing refers* to the number of contributors attached in carbon copy. The two effects are both transformed into dummy variables by choosing as cut-off the upper quartile of their distribution. The two dummies are included to control for the effect of extra saliency of certain software bugs on the propensity to attract the attention of contributors. A summary of exogenous actor-specific covariates is shown in Table 3.

Table 1.1: Summary of *attentional mechanisms of theoretical interest*

Assortative attention spread	*Preferential attachment* - *or* tendency for active contributors to engage with popular software bugs		$\sum_{p} y_{ip} \sqrt{\sum_{p} y_{ip}} \sqrt{\sum_{i} y_{ip}}$
Attention clustering	*Four-cycle* - or tendency for pairs of contributors to engage with the same software bugs		$\sum_{i,j} y_{ip1} y_{ip2} y_{jp1} y_{jp2}$

Note. Blue squares represent software contributors. Red circles represent software bugs. Solid lines represent existing ties. Dashed lines in light gray represent potential ties. y_{ip} indicates a tie between contributor *i* and software bug *p*.

Table 2.2: Summary of endogenous network controls.

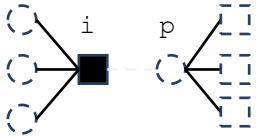
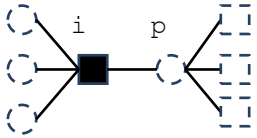
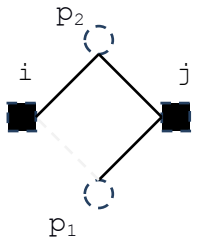
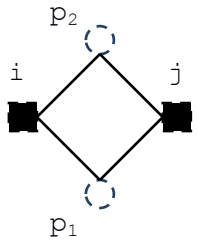
Parameter	Included to control for	Configuration(t_i)	Configuration(t_{i+1})	Network statistics
Outdegree (density)	Overall tendency to engage with a software bug			$\sum_p y_{ip}$
Contributor activity	*Positive feedback in activity* - or tendency for active contributors to engage with a progressively larger number of software bugs			$\sum_p y_{ip} \sqrt{\sum_p y_{ip}}$
Software bug popularity	Tendency for popular software bugs to attract extra contributors			$\sum_p y_{ip} \sqrt{\sum_i y_{ip}}$

Note. Blue squares represent software contributors. Red circles represent software bugs. Solid lines represent existing ties. Dashed lines represent potential ties. Y_{ip} indicates a tie between contributor *i* and software bug *p*.

Table 2.3: Summary table for exogenous actor-specific covariates (contributors = 135, software bugs = 719)

Attribute	Type	Motivation	Operationalization	Proportion
Contributor experience	Constant	Controls for the effect of learning	One if a contributor was active in the prior release cycle, zero otherwise	0.430
Core developer	Constant	Controls for the effect of formal roles	One if a contributor is classified as "core developer," zero otherwise	0.029
Contributor CC'ing only	Changing	Controls for the effect of marginal roles	One if a contributor was involved only in cc'ing activities, zero otherwise	0.155(t_1) 0.141(t_2) 0.133(t_3)
Software bug severity	Constant	Controls for the effect of centralizing tendencies	One if a software bug had a severity level above "normal" in the prior release cycle, zero otherwise	0.349
Software bug communication	Changing	Controls for the effect of awareness in the project	One if a software bug lies in the upper quartile of the distribution of number of received comments	0.168(t_1) 0.227(t_2) 0.191(t_3)
Software bug CC'ing	Changing	Controls for the effect of centralizing tendencies	One if a software bug lies in the upper quartile of the distribution of number of contributors assigned to software bugs in carbon copy	0.136(t_1) 0.153(t_2) 0.129(t_3)

Table 2.4: Summary of actor-relations interaction terms

Parameter	Included to control for	Configuration(t_i)	Configuration(t_{i+1})	Network statistics
Experienced contributors attention spread	Tendency for active experienced contributors to engage with popular software bugs			$\sum_p y_{ip} v_i \sqrt{\sum_p y_{ip}} \sqrt{\sum_i y_{ip}}$
Experienced contributors attention clustering	Tendency for pairs of experienced contributors to engage with the same software bugs			$\sum_{i,p} y_{ip1} v_i y_{ip2} y_{jp1} y_{jp2}$

Note. Black squares (circles) represent contributors (software bugs) with an attribute. Dotted squares (circles) represent contributors (software bugs) without that same attribute. Solid lines represent existing ties. Dashed lines represent potential ties. y_{ip} indicates a tie between contributor *i* and software bug *p*. v_i indicates that contributor *i* is an experienced contributor whereas v_p indicates that software bug *p* is a severe bug. Therefore $y_{ip1}v_i$ indicates a tie between experienced contributor *i* and software bug p_1

Finally, the interaction of actor-specific covariates with structural network effects are included to capture the tendency of nodes with specific attributes to become part of specific local network structures. I include in the model two interaction terms to test for H3a and H3b in an attempt to test whether the effect of embeddedness in attention sub-network structures varies according to individual characteristics of organizational members. *Experienced contributors attention clustering* and *Experienced contributors attention spread* are included to control for the specific tendency of experienced contributors, respectively, to be part of local collaboration clusters and to be attracted to problems that have received an exceptional level of attention. A summary of interaction terms included in our model is shown in Table 4.

2.4.3 Stochastic Actor-oriented models for bipartite data

I start by assuming that the network structure observed at any one time develops as a result of interdependent individual decisions. This is the main assumption underlying a newly derived class of Stochastic Actor-oriented Models (SAOM) for social networks (Snijders, van de Bunt & Steglich, 2010). Actors are only allowed to change the ties under their direct control (i.e., values in their own row), and no single actor has control over the entire network structure. Statistically, this assumption leads to a representation of the network structure that is observed at any moment as a realization of a continuous-time Markov Chain Y(t) – where observed realizations are $y(t_\tau)$ (with $\tau = 1,2,\ldots,T$) (Snijders, 2001). At any point in time the process produces the observed network Y(t) = y. In this specific case I discuss y as a bipartite network of size *NxM*, with tie variables $y_{ip} = 1$ if participant *i* engages with problem *p*, and $y_{ip} = 0$ otherwise. Formally, the model is a continuous-time Markov process, whose state space is defined in terms of all the possible combinations of network ties (Snijders, 2001).

Linking SAOM to data requires specification of two main components. The first is a *rate function* $\lambda_i(\alpha, y)$ which controls how quickly opportunities for changing network ties arise. In our case, the relevant decision concerns the change in the individual portfolio of network ties to problems. Participants get opportunities to change their affiliation to problems at the rate:

$$\lambda(Y(t)) = \exp\left(\alpha_0 + \sum_k \alpha_k x_{ik}\right). \quad (1)$$

The rate may be constant between observation moments (when $\alpha_k = 0$, for all *k*), or it may change depending on actor-specific covariates(x_{ik}).In the model I estimate below I make the number of opportunities for change dependent on differences in the status of participants (defined in terms of "core" versus "non-core" status). In this way I allow for the possibility that core developers enjoy greater freedom to change their affiliations to problems over time.

The second component of SAOM is the individual decision of participants (rows) to change their affiliation to problems (columns). This decision is controlled by an *evaluation function (f_i)* representing the relative attractiveness for participant *i* of moving from *y* to *y'*, where *y* and *y'* are successive network configurations differing in terms of only one tie. Among the possible *m* changes that a participant can make at any one time, he or she is assumed to choose so that $f_i(y, y', x) + e(y, y', x)$ is maximized. In this formulation f_i is a deterministic evaluation function, *x* is a set of covariates, and *e* is a random disturbance or "error term." The deterministic part of the evaluation function assumes the typical linear form (Snijders, van de Bunt & Steglich, 2010)

$$f_i(\beta, y) = \sum_k \beta_k s_{ik}(y), \quad (5)$$

where $f_i(\beta, y)$ is the value of the evaluation function for participant *i* depending on the state of the network (*y*), and the term $s_{ik}(y)$ are the "effects" which may be associated with: (i) attention mechanisms; (ii) actor-specific covariates representing "exogenous" characteristics of the participants

or problems, and (iii) interactions between network *motifs* and exogenous covariates. Finally β_k are parameters that may be estimated from data representing the weight, or strength, of the corresponding effect.

Individual decisions to change network ties are based on a comparison of the values of the evaluation function computed across the permissible choice options. More specifically, if *y'=y(i→p)* denotes the network that would be observed if participant *i* changed his connections to problem *p* (creation of a new tie or termination of an existing tie, dependent on the existence of a tie), then the probability ofobserving one change would be:

$$\Pr(y(i \to p)|x) = \frac{\exp\left(\sum_k \beta_k s_{ik}(y(i \to p), x\right)}{\sum_h \exp\left(\sum_k \beta_k s_{ik}\left(y(i \to h), x\right)\right)}. \quad (6)$$

With additional distributional assumptions the model just described is consistent with revealed preference interpretations as described, for example, in Maddala (1983: Chapter 3) and established by McFadden (1974). According to this interpretation, if participant *i* changes his affiliation profile (his row in the bipartite network), producing a change from configuration *y* to *y',* then he is acting as if he *prefers y' to y.*

2.4.4 Model estimation and evaluation

Because model estimation involves computing the transition probabilities between all possible neighboring states of the network, the state space of the model consists of all possible trajectories linking network configurations observed in adjacent time periods. Under such conditions conventional statistical estimation is unfeasible but parameters estimates may be obtained via an iterative Robbins-Monro (1951) stochastic approximation algorithm using the Method of Moments (MoM) estimation

procedure (Snijders, 2001). The algorithm computes the evolution process of the network multiple times via Markov Chain Monte Carlo (MCMC) simulations implied by the current model specification (Snijders, 2002). Parameter values are updated after each simulation reflecting the deviations between generated and observed statistics corresponding to the effects included in the model (Schweinberger & Snijders, 2007). For all estimated parameters averages and standard errors of these deviations are computed across all the simulated networks and used to compute *convergence t-ratios*. Convergence *t*-ratios for all the estimates I report in Table 6 are smaller (in absolute value) than 0.1 – the critical value below which a model is considered to be fully convergent (Snijders et al., 2010). As suggested by Ripley et al. (2011), I repeated such estimation 3000 times to obtain reliable estimates of the standard errors.

2.5 Results

The discussion of the empirical results is organized around Table 2.5. Convergence t-ratios for all reported estimates are smaller than 0.1 in absolute values. The null model (Model 1), accounts only for the overall propensity of contributors to engage with software bugs, and for the effects of exogenous actor-specific covariates on such propensity. The main effects of the attentional mechanisms of theoretical interest on the individual choice of contributors (accounting for H1 and H2) are added in our main-effects model (Model 2). Finally our full model (Model 3) adds interaction terms to Model 2 which account for the moderating effects of organizational members' tenure on the patterns of attention allocation (H3a and H3b).

Table2.5: Method of Moment estimates of SAOM for bipartite networks (Estimated standard errors in parentheses).

	Effects	Model 1	Model 2	Model 3
Change rate effects	Rate of network change 1	2.548 (0.171)	3.453 (0.390)	3.780 (0.450)
	Rate of network change 2	1.876 (0.129)	1.870 (0.130)	2.001 (0.151)
	Rate of network change 3	4.159 (0.216)	3.934 (0.301)	3.971 (0.451)
	Core developer effect on rate	3.221*** (0.084)	3.386*** (0.081)	3.301*** (0.950)
Exogenous actor-specific covariates	Software bug communication	-0.770*** (0.121)	-0.860*** (0.127)	-0.821*** (0.126)
	Software bug severity	-0.270*** (0.065)	-0.317*** (0.069)	-0.366*** (0.069)
	Software bug CC'ing	0.399** (0.124)	0.637*** (0.131)	0.632*** (0.130)
	Core developer	1.559*** (0.103)	-0.658*** (0.161)	-0.941*** (0.221)
	Contributor CC'ing only	-0.609*** (0.124)	-0.255* (0.113)	-0.270* (0.121)
	Contributor experience	0.959*** (0.094)	0.769*** (0.103)	0.538** (0.157)
Endogenous network effects	Outdegree	-2.498*** (0.072)	-3.597*** (0.297)	-3.275*** (0.347)
	Software bug popularity		-0.048 (0.183)	-0.493* (0.226)
	Contributor activity		0.453*** (0.049)	0.442*** (0.056)
Attentional mechanisms of theoretical interest	**Attention Clustering**		0.033* (0.016)	0.056 (0.047)
	Attention Spread		-0.005* (0.002)	-0.019 (0.020)
Interactions of theoretical interest	**Contributor experience * clustering**			0.142* (0.065)
	Contributor experience * attention spread			-0.041* (0.016)

The *Rates of network change (1-3)* provide an estimate of the average number of opportunities for changing individual affiliation to problems. In the full model contributors face on average

approximately 4 opportunities for change in the last time period. Across all models choice opportunities appear to decrease in the middle of the release cycle. As expected, core developers enjoy significantly more choice opportunities than all other contributors throughout the release cycle, as captured by the strongly positive and significant *Core developer* effect on rate which implies an average of 3 extra opportunities. The negative *Outdegree* parameter indicates that attempting to resolve a software problem is a costly action. This result suggests that attention is a scarce resource. Problems of attention allocation arise as a consequence.

In Model 1 I include exogenous contributor-specific as well as problem-specific covariates. The negative and significant parameter estimate for *Software bug communication* indicates that software bugs generating more discussion also tend to "scare away" contributors. Specifically, the odds of a software bug generating an exceptional level of discussion being selected by a contributor against not being selected are [e^(-0.770)=] 0.464, meaning that bugs generating long discussions are 53.6% less likely to be engaged with than bugs stimulating a lower level of interest. Similarly, the negative and significant parameter for *Software bug severity* reveals that more severe bugs tend to discourage contributors from taking action – the respective odds being 0.763, or a decrease by 23.7%. This could be due to the fact that only a restricted number of contributors have sufficient confidence and skills to attend to software bugs collectively classified as severe. As a consequence, action directed towards severe bugs is relatively rare. By contrast, the number of contributors choosing to allocate their attention on specific software bugs by putting themselves in carbon copy of the relative bug reports increases the attractiveness of those software bugs, with a positive and significant *Software bug CC'ing* effect – the associated odds being 1.49, or an increase by 49%. However, the negative and significant parameter for *Contributor CC'ing only* indicates that, *ceteris paribus*, contributors only putting themselves in carbon copy of bug reports tend to produce fewer problem-solving attempts. Finally, the

positive and significant parameter estimates for *Contributor experience* and *Core developer* indicate, respectively, that experienced contributors and core developers tend to be more active. Specifically, the odds of experienced contributors engaging with a software bug against not engaging with it are 2.6 times as large as the odds for inexperienced contributors.

In Model 2 I include local dependencies taking the form of the various endogenous attention mechanisms discussed earlier in this paper. The significant parameter estimate for *Attention clustering* supports Hypothesis 1, indicating that contributors tend to develop patterns of repeated collaboration: collaborating on a problem makes it more likely that two contributors will collaborate again on future problems. The negative and significant *Assortative attention spread* indicates that very active contributors choose to work on problems that are not already attended to by other participants: this represents strong evidence against preferential attachment and supports Hypothesis 2. In addition, the positive and significant parameter estimate for *Contributor activity* shows evidence of reinforcing feedback: the higher the level of current activity, the higher the level of future activity. One possible interpretation of this effect is as a consequence of processes of "learning by doing" which progressively lowers the cost of individual units of attention. The non-significant parameter for *Software bug popularity* indicates that already popular software bugs are not more likely to attract attention additional in the future. As a consequence of the associated tendency toward decentralized problem-solving, attention is more homogenously distributed across problems. Finally, parameters estimates for exogenous contributor-specific covariates in Model 2 reveal that experienced contributors do not exhibit a significantly different choice behavior once I control for the main effect of the attentional mechanisms.

The inclusion of interaction terms in Model 3 does not alter the parameter estimates for exogenous covariates already included in the previous model, with the exception of *Core developer*. In Model 3 experienced contributors do show higher propensity to attempt resolution of software bugs,

thus suggesting that a positive effect of experiential learning might influence the association between problems and participants (Argote & Epple, 1990) over and above the effect of attentional mechanisms taken into account.

Contributor experience moderates the effects of the attentional mechanisms of theoretical interest in the hypothesized directions. Together with the no longer significant *clustering* effect, the significant and positive *Experienced contributors * clustering* supports H3a, indicating that patterns of sustained collaboration when attempting to resolve software problems are driven by cognitive inertia developed by experienced contributors. Likewise, considered together with non-significant *assortative attention spread,* the significant and negative *Experience * attention spread* effect supports H3b and indicates that only experienced contributors prefer to select problems that are not already engaged with by other participants. The consequence is a tendency towards an anti-hierarchical pattern in which the most popular problems are "decoupled" from the most active – and experienced – participants.

2.6 Discussion and conclusions

In this paper I tackled the relevance of attention networks in understanding processes of problem-solving in open productions. Following the tradition of the Carnegie School (Simon, 1947: March & Simon, 1958: Cyert & March, 1963) I based my arguments on the idea that organizational attention is a valuable and scarce resource. Stemming from this idea is the evidence that problems compete for the limited attention of problem-solvers. The ABV literature (Simon, 1947; Ocasio, 1997) posits that organizations channel the attention of their participants through the design of working routines and communication structures. I extended this view by presenting two attentional mechanisms – attention clustering and attention spread – that show how apparently individualistic attention allocation decisions are in fact the result of the embeddedness of individuals in a complex network of

contextual interdependencies linking problems and problem-solvers in an organization.

The analysis suggests that problems show a tendency towards a disassortative attention spread, indicating that already popular software bugs are not more likely to attract additional attention in the future by active individuals. As a consequence of the associated tendency toward decentralized problem-solving, attention is more homogenously distributed across problems. This result is in line with recent studies on network configurations in decentralized, virtual projects (Faraj & Johnson, 2011; Conaldi & Lomi, 2012), where a more homogeneous attention spread facilitates the recognition and resolution of the highest number of problems, which in turn guarantees the long-term viability of the project. In particular, the analysis shows a positive tendency towards attention clustering, indicating that attentional processes tend to follow repetitive patterns of joint interest towards problem-solving: collaborating on a problem makes it more likely that two contributors will allocate their joint attention to future problems. Furthermore, the interaction of organizational experience with the two network configurations seem to suggest that experienced members are responsible for driving the tendency towards attention clustering around already familiar problems, due to the effect of reinforcing cognitive schemas. On the other hand, organizational experience also drives the preference away from preferential attachment, showing that even though experienced members are less likely to allocate their attention outside their familiar cluster of problems, they try to boost or maintain their reputation by picking less obvious and less popular problems.

My modeling efforts are consistent with March and Olsen's (1976) view that the attention allocation of one decision-maker is a function of the attention of neighboring decision-makers. They are also in line with T.C. Schelling's illuminating intuition that: "people's behavior depends on how many are behaving a particular way, or how much they are behaving that way" (1978: p. 94). As is the case in many of Shelling's examples in our case individual decisions generate endogenously transparent and freely accessible information to which potential participants may pay attention in

determining their own level of contribution to collective production efforts. Finally, this paper extends recent work on distributed social cognition (Smith & Collins, 2009; Kaplan, 2011) where market valuations are not performed in a vacuum, but are embedded in attention networks linking interdependent actors' views (Prato & Stark, 2013). What I have added to this perspective is a series of specific mechanisms through which individual acts of attention allocation concatenate and reproduce behavioral patterns that are used to produce collective problem-solving efforts.

Two main limitations of this work deserve special attention as they reveal clear opportunities for further research. The first limitation is inherent in the focus on differences in individual decisions to allocate attention to organizational problem-solving activities. Our emphasis on individual decisions and effort precluded analysis of the outcomes of such decisions. As a consequence I was unable to assess, for example, how effective the resolution of problems actually was during the release cycle of the software, how long on average problems remained unresolved within the project (the problem latency time), or how durable (or stable) the solutions that contributors implemented were (Cohen et al., 1972). Addressing these issues requires a research design oriented toward the consequences of problem-solving behavior. In this paper I focused on its antecedents i.e., on the individual decisions that affiliate organizational members to organizational problems. Data necessary to examine the effectiveness of observed problem-solving attempts are available. These may serve to illuminate the network dependency of problem-solving effectiveness by allowing analysis of the embeddedness of individual behavior and the outcomes of that behavior within the attention networks that enable and constrain organizational problem-solving.

The second limitation relates to the empirical scope of this research. The results are not based on a random sample of F/OSS productions which might be considered representative of the open productions field, but instead constitute a single case study of a F/OSS project, albeit one which is relatively large and successful, and which has been analyzed using longitudinal data. As such the

generalizability of the results reported here may only be assessed through systematic replication. .

Despite these limitations, I believe this study advances the debate on the structural perspective on attention and situated cognition. It brings to bear different attention allocation mechanisms defined by the relationships between problems and participants within the context of open productions, which give rise to interdependent patterns of problem-solving activity.

References

Bailey, D. E., Leonardi, P. M., & Barley, S. R., (2012). The lure of the virtual. *Organization Science*, 23(5): 1485-1504.

Barabási, A.L., & Albert, R., (1999). Emergence of scaling in random networks. *Science*, 286: 509–512.

Barnett, M. L. (2008). An attention-based view of real options reasoning. *Academy of Management Review, 33*(3), 606-628.

Bouquet, C., & J. Birkinshaw., (2008). Weight versus voice: How foreign subsidiaries gain attention from corporate headquarters. *Acad. Management J*, 51(3): 577–601.

Breiger, R.L., (1974). The duality of persons and groups. *Social Force,s* 53 (2), 181–190.

Breiger, R., (2000). A toolkit for practice theory. *Poetics*, 27 (2–3): 91–115.

Breiger, R. L., & Mohr, J. W., (2004). Institutional logics from the aggregation of organizational networks: Operational procedures for the analysis of counted data. *Computational & Mathematical Organization Theory*, *10*(1): 17-43.

Burt, R. S., (2004). Structural holes and good ideas. *American Journal of Sociology*, 110(2): 349-399.

Carley, K., (1991). A theory of group stability. *American Sociological Review*, 331-354.

Cho, T. S., & Hambrick, D. C., (2006). Attention as the mediator between top management team characteristics and strategic change: The case of airline deregulation. *Organization Science*, *17*(4): 453-469.

Cohen, M. & P. Bacdayan., (1994). Organizational routines are stored as procedural memory: evidence from a laboratory study. *Organization Science*, 5(4)\: 554–568.

Cohen, M. D., March, J. G., & Olsen, J. P., (1972). A garbage can model of organizational choice. *Administrative science quarterly*, 1-25.

Conaldi, G., & Lomi, A., (2013). The dual network structure of organizational problem solving: A case study on Open Source Software development. *Social Networks*.

Conaldi, G., Lomi, A., & Tonellato, M., (2012). Dynamic models of affiliation and the network structure of problem solving in an Open Source Software project. *Organizational Research Methods*, 15(3): 385-412.

Corbetta, M., & G. L. Shulman., (2002). Control of goal-directed and stimulus-driven attention in the brain. *Nature Rev. Neurosci*, 3(3): 201–215.

Crowston, K., Howison, J., (2005). The social structure of free and open source software development. *First Monday*, 10 (2).

Crowston, K., Wei, K., Li, Q., Howison, J., (2006). Core and periphery in Free/Libre and Open Source software team communications. *Institute for Software Research*, Paper 489 http://repository.cmu.edu/isr/489

Crowston, K., & Scozzi, B., (2008). Bug fixing practices within Free/Libre Open Source Software development teams. *Journal of Database Management*, 19(2): 1-30.

Crowston, K., (2008\0. The bug fixing process in proprietary and Free/Libre Open Source Software: a coordination theory analysis. In: Grover, V., Markus, M.L. (Eds.), *Business Process Transformation: Advances in Management Information Systems*. M.E. Sharpe, Armonk, NY.

Cyert, R. M., & March, J. G., (1963). *A behavioral theory of the firm*. Englewood Cliffs, NJ.

Dabbish, L., Stuart, C., Tsay, J., & Herbsleb, J., (2012, February). Social coding in github: transparency and collaboration in an open software repository. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work*, (pp. 1277-1286). ACM.

Dahlander, L., & O'Mahony, S., (2011). Progressing to the center: Coordinating project work. *Organization Science*, *22*(4): 961-979.

Easley, D., Kleinberg, J., (2010). *Networks, Crowds, and Markets: Reasoning About a Highly Connected World*. Cambridge University Press, New York, NY.

Eisenhardt, K,. (1989). Making fast strategic decisions in high-velocity environments. *Academy of Management Journal*, 32(3): 543-576.

Faraj, S., & Johnson, S. L., (2011). Network exchange patterns in online communities. *Organization Science*, *22*(6): 1464-1480.

Festinger, L. (1962). *A theory of cognitive dissonance*. Stanford university press.

Fligstein, N., 2002 The Architecture of Markets: An Economic Sociology of Twenty-First-Century Capitalist Societies. Princeton, NJ: Princeton University Press.

Goldberg, A. 2011. Mapping shared understandings using relational class analysis: The case of the cultural omnivore reexamined. American Journal of Sociology, 116(5): 1397-1436.

Guo, P. J., Zimmermann, T., Nagappan, N., & Murphy, B. (2010, May). Characterizing and predicting which bugs get fixed: An empirical study of Microsoft Windows. In *Software Engineering, 2010 ACM/IEEE 32nd International Conference on* (Vol. 1, pp. 495-504). IEEE.

Hansen, M. T., M. R. Haas. 2001. Competing for attention in knowledge markets: Electronic document dissemination in a management consulting company. Admin. Sci. Quart. 46(1) 1–28.

Hoffman, A. J., & Ocasio, W. (2001). Not all events are attended equally: Toward a middle-range theory of industry attention to external events. *Organization Science, 12*(4), 414-434.

Hsu, G. 2006. Jacks of all trades and masters of none: Audiences' reactions to spanning genres in feature film production. *Administrative Science Quarterly, 51*: 420-450.

Jeppesen, L. B. & Lakhani, K. R. 2010. Marginality and problem-solving effectiveness in broadcast search. *Organization Science, 21(5)*: 1016-1033.

Kaplan, S. 2011. Research in Cognition and Strategy: Reflections on Two Decades of Progress and a Look to the Future. *Journal of Management Studies, 48*: 665-695

Klimoski, R., & Mohammed, S. (1994). Team mental model: Construct or metaphor?. *Journal of management*, *20*(2), 403-437.

Koskinen, J.,Edling, C., 2013. Modelling the evolution of a bipartite network: Peer referral in interlocking directorates. *Social Networks*. In press.

Lerner, J., & Tirole, J. (2002). Some simple economics of open source. *The journal of industrial economics*, *50*(2), 197-234.

Levinthal, D. A. & March, J. G. 1993. The myopia of learning. Strategic Management Journal, 14(8): 95.

Liang, D. W., Moreland, R., & Argote, L. (1995). Group versus individual training and group performance: The mediating role of transactive memory. *Personality and Social Psychology Bulletin*, *21*(4), 384-393.

Lounsbury, M. (2007). A Tale of Two Cities: Competing Logics and Practice Variation in the Professionalizing of Mutual Funds. *Academy of Management Journal*, *50*(2), 289-307.

Maddala, G.S., 1983. Limited-Dependent and Qualitative Variables in Econometrics. Cambridge University Press.

March, J. G. 1991. Exploration and Exploitation in Organizational Learning. Organization Science, 2: 71-87.

March, J. G., H. Simon. 1958. Organizations. John Wiley & Sons, New York.

March, J. G., & Olsen, J. P. (1976). *Ambiguity and choice in organizations.*

McFadden, D., 1974. Conditional logit analysis of qualitative choice behavior. In: Zarembka, P. Editor, Frontiers in Econometrics, Academic Press: New York, NY, 105–142.

Michlmayr, M., 2007. Quality improvement in volunteer free and open source software projects: exploring the impact of release management. PhD. dissertation. Centre for Technology Management, Institute for Manufacturing, University of Cambridge (U.K).

Nelson, R. R., & S. Winter 1982 An Evolutionary Theory of Economic Change. Cambridge, MA: Harvard University Press.

Newman, M.E.J., 2003. Mixing patterns in networks. Phys. Rev. E67, 026126.

Newman, M.E.J., Park, J., 2003. Why social networks are different from other types of networks. Phys. Rev. E 68, 036122.

Ocasio, W. 1997. Towards an attention-based view of the firm. Strategic Management J. 18(S1) 187–206.

Ocasio, W. 2011. Attention to attention. Organization Science, 22(5): 1286-96.

Ocasio, W., J. Joseph. 2005. An attention-based theory of strategy formulation: Linking micro- and macroperspectives in strategy process. Adv. Strategic Management 22 39–61.

Pattison, P. E., & Breiger, R. L. (2002). Lattices and dimensional representations: matrix decompositions and ordering structures. *Social Networks*, *24*(4), 423-444.

Pattison, P.E., Robins, G.L., 2002. Neighborhood-based models for social networks. Sociological Methodology 32, 301–337.

Powell, W. W., Koput, K. W., & Smith-Doerr, L. (1996). Interorganizational collaboration and the locus of innovation: Networks of learning in biotechnology. *Administrative science quarterly*, 116-145.

Prato, M., & Stark, D. (2013, January). Peripheral Vision in Financial Markets: How attention networks shape valuation. In *Academy of Management Proceedings* (Vol. 2013, No. 1, p. 15923). Academy of Management.

Robbins, H., Monro, S., 1951. A stochastic approximation method. Annals of Mathematical Statistics 22, 400–407.

Robles, G., González-Barahona, J.M., Izquierdo-Cortazar, D., Herraiz, I., 2009. Tools for the study of the usual data sources found in libre software projects. International Journal of Open Source Software and Processes 1 (1), 24–45.

Schelling, T.C., 1978. Micromotives and Macrobehavior. New York: WW Norton and Company.

Schweinberger, M., Snijders, T.A.B., 2007. Markov models for digraph panel data: Monte Carlo-based derivative estimation. Computational Statistics and Data Analysis 51 (9), 4465–4483.

Simon, H. A. 1947. Administrative Behavior: A Study of Decision- Making Processes in Administrative Organizations. Macmillan, Chicago.

Smith, E. R., & Collins, E. C. (2009). Contextualizing person perception: Distributed social cognition. *Psychological Review, 116*: 343–364.

Snijders, T.A.B., 2001. The statistical evaluation of social network dynamics. Sociological Method 31, 361–395.

Snijders, T.A.B., 2002. Markov chain Monte Carlo estimation of exponential random graph models. Journal of Social Structure 3 (2).

Snijders, T.A.B., van de Bunt, G.G., Steglich, C.E.G., 2010. Introduction to stochastic actor-based models for network dynamics. Social Networks 32 (1), 44–60.

Stasser, G., & Titus, W. (1985). Pooling of unshared information in group decision making: Biased information sampling during discussion. *Journal of personality and social psychology*, *48*(6), 1467.

Stuart, H. C., Dabbish, L., Kiesler, S., Kinnaird, P., & Kang, R. (2012, February). Social transparency in networked information exchange: a theoretical framework. In Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work (pp. 451-460). ACM.

Sullivan, B. N. 2010. Competition and beyond: Problems and attention allocation in organizational rulemaking processes. Organ. Sci. 21(2) 432–450.

Stuart, T. & Podolny, J. 1996. Local search and the evolution of technological capabilities. Strategic Management Journal, 17(1): 21-38.

Thompson, J. D. (1967). *Organizations in action: Social science bases of administrative theory*. Transaction Pub.

Thornton, P. H., & Ocasio, W. (1999). Institutional logics and the historical contingency of power in organizations: Executive succession in the higher education publishing industry, 1958-1990. *American Journal of Sociology*, *105*(3), 801-843.

Tsai, W., Su, K., & Chen, M.J. 2011. Seeing Through the Eyes of a Rival: Competitor Acumen Based on Rival-Centric Perceptions. *Academy of Management Journal, 54*: 761–778.

von Krogh, G., Haefliger, S., Spaeth, S., & Wallin, M. W. (2012). Carrots and rainbows: Motivation and social practice in open source software development. *MIS Quarterly-Management Information Systems*, *36*(2), 649.

Wang, P., Robins, G.L., Pattison, P.E., 2012. Exponential Random Graph Model specifications for bipartite networks: A dependence hierarchy. Social Networks. In press.

Williams, C., & Mitchell, W. 2004. Focusing firm evolution: The impact of information infrastructure on market entry by US telecommunications companies, 1984-1998. Management Science, 50(11): 1561-1575.

Winter, S. G., Cattani, G., & Dorsch, A. (2007). The value of moderate obsession: Insights from a new model of organizational search. *Organization Science*, *18*(3), 403-419.

Woods, D. D., Patterson, E. S., & Roth, E. M. 2002. Can we ever escape from data overload? A cognitive systems diagnosis. Cognition, Technology & Work, 4(1): 22-36.

Yu, J., Engleman, R. M., & Van de Ven, A. H. (2005). The integration journey: An attention-based view of the merger and acquisition integration process. *Organization studies*, *26*(10), 1501-1528.

Zanetti, M. S., Scholtes, I., Tessone, C. J., & Schweitzer, F. (2013). Categorizing Bugs with Social Networks: A Case Study on Four Open Source Software Communities.

Zuckerman, E. W. 1999. The categorical imperative: Securities analysts and the illegitimacy discount. American Journal of Sociology, 104(5): 1398-1438.

Zuckerman, E. W. 2004. Structural incoherence and stock market activity. *American Sociological Review*, 405–432.

Chapter 3

THE EFFECT OF EXPERTISE DIVERSITY ON GROUP LEARNING AND PERFORMANCE: A CASE STUDY IN OPEN SOURCE SOFTWARE[1]

Abstract

Repeated collaboration represents an important mechanism through which organizations learn. Research on group learning has shown that experience of working together helps groups within organizations achieve better coordination, establish shared norms and reduce task completion time. However, sustained collaboration of a group over a long period of time may induce group members to focus on existing solutions and habitual routines, and discourage efforts to seek further afield for new and better approaches to problem solving. Building on this evidence, I argue that group learning is contingent on the distribution of task-related expertise among group members. Introducing the moderating effect of task-related expertise diversity on group learning and performance, I examine the decentralized resolution efforts of 2143 individuals on 3470 software bugs in a large Free/Open Source Software project (F/OSS) over a period of 10 years. I find that groups that are composed of individuals with homogeneous expertise outperform groups with heterogeneous expertise at lower levels of prior collaboration, because I argue that homogeneity helps establish common ground in the early phases of collaborative practice. However, at higher levels of prior collaboration groups composed of individuals with heterogeneous expertise outperform groups with homogeneous expertise, because I argue that specialization and expertise diversity are essential to establish a clear understanding of "who knows what". I discuss the implications and contribution of my work within the broader contexts of research on group dynamics and organizational learning.

[1] with Guido Conaldi

3.1 Introduction

Recent years have witnessed an increase in the prominence of open productions as some industries shift toward systems of decentralized collaboration in which participants are willing to bear private costs in order to provide public goods (Levine & Prietula, 2014; Baldwin & von Hippel, 2011). Open productions usually take the form of project-based organizations composed of temporary groups that combine several individuals to carry out collective, short-term tasks (Schwab & Miner, 2008; Powell, 2003). Groups in project-based organizations stay together only as long as they work on one task. Their members may leave or join the group during the project lifetime to participate in the resolution of other tasks (Huckman, Staats & Upton, 2009).

The main purpose of organizational groups is to let members with heterogeneous expertise interact to allow the performance of complex tasks that could not be undertaken individually (Casciaro, 2013). There is a long-standing stream of research in organization theory and design examining the coordination mechanisms needed to integrate the complementary – and possibly diverse – knowledge of participants in such organizational groups (Mintzberg, 1979; Thompson, 1967). Repeated collaboration between organizational team members over the resolution of shared problems has been shown to be a pivotal mechanism for such knowledge integration and the successful execution of interdependent tasks (Skilton & Dooley, 2010).

Research has extensively shown that groups learn from members' repeated collaboration (Hinds, Carley, Krackhardt & Wholey, 2000; Faulkner, 1983). For example, experience of working together help teams reduce procedure completion time (Reagans, Argote & Brooks, 2005), establish trustworthy norms (Uzzi, 1996), achieve better coordination (Faraj & Sproull, 2000), and promote the sharing of tacit knowledge (Uzzi & Lancaster, 2003). Albeit the literature on learning has provided extensive support to the positive effect of repeated collaboration on group performance, a competing

stream of literature emphasizes dysfunctional dynamics in repeated collaboration. Katz (1982) examined the effect of group longevity on performance and found that groups whose members exhibited very high levels of repeated collaboration tend to underperform due to decreased communication rates among group members. Likewise, Perretti and Negro (2007) reported a negative relationship between prior collaboration and project performance in the Hollywood film industry, revealing the existence of a phenomenon that was subsequently found to have similarly detrimental effects in other creative industries (Skilton & Dooley, 2010).

This paper aims to shed light on the contested relationship between group members' repeated collaborations and group performance in the context of open productions. I do this by examining the theoretical foundations of existing mixed findings and indicate more precisely the contingent factors for understanding under what conditions higher repeated collaboration is likely to be beneficial or detrimental for project group performance. In particular, I believe that lack of researchers' attention to the distribution of task-related expertise among group members is a determining factor in the failure to thus far unravel this apparent inconsistency in findings. Existing research has shown that groups composed of members with a lower degree of expertise diversity tend to share more common ground (Stasser & Titus, 1985; Kraut, Fussell, Brennan & Siegel, 2002) and thus develop a higher absorptive capacity which is essential to ease knowledge transfer (Cohen & Levinthal, 1990). Meanwhile, a higher degree of expertise diversity amongst members of a task group tends to facilitate the recognition of expertise specialization, which is essential for the development of a shared understanding of "who knows what" in the team (Liang, Moreland & Argote, 1995).

My study extends this earlier research in two specific ways. Firstly, I show that the relationship between group experience of working together and group performance is non-monotonic, with a state of decreasing returns coming into force beyond a certain point. Secondly, I demonstrate that expertise diversity is a moderator of the relation between repeated collaboration and team performance. Groups

composed of individuals with homogeneous expertise perform better than groups with heterogeneous expertise at lower levels of prior collaboration, because homogeneity helps establish common ground in the early stages of a group project (Kraut et al., 2002). At the other end, groups that are composed of individuals with heterogeneous expertise perform better than groups with homogeneous expertise at higher levels of prior collaboration. In this latter scenario specialization and expertise diversity are seen as crucial for maintaining group performance in the long run (Kane, Argote & Levine, 2005).

The empirical part of the study uses data I collected on problem-solving tasks observed within Free/Open Source Software (F/OSS) projects. In particular, I examine bug-fixing activities in one of the largest and most successful F/OSS projects to date, the Apache web server. The emergence of open, collaborative forms of economic production such as Free/Open Source Software projects provides an optimal setting for testing my hypotheses. Theories of "social transparency" (Dabbish et al., 2012), associated with the diffusion of open productions, refer to the visibility and traceability of the complete history of actions taken by a distributed workforce in the task environment. In Free/Open Source Software projects developers are free to decide which task to perform, and each action is then coded in a knowledge repository which is transparent to any other developer. Exogenous coordination mechanisms such as hierarchy or planning play a limited role. Contributors organize their production on the basis of information generated by the previous actions of fellow participants (Stuart et al., 2012). When engaging in collaborative problem-solving activities organizational members learn directly from the collective experience while forming fluid working groups around shared artifacts, which store transparent information about previous actions of other members. As a consequence, this setting allows us to directly observe the emergence of repeated collaboration and knowledge integration from observable work practices, and to rule out possible alternative explanations related to formal team structure which have been investigated in other recent studies (Bunderson & Boumgarden, 2010).

The study is organized as follows: first I review the relevant literature that addresses the issues

of collaboration and knowledge integration in decentralized, self-managed groups. Then I develop my hypotheses on how repeated collaboration affects group performance and on how expertise diversity moderates this relationship. After describing the data and the statistical model I use to test the theory, I report my findings. I conclude with a discussion of how these findings contribute to the literature on organizational learning.

3.2 Theory and hypotheses

Organizational scholars have widely recognized the relevance of groups as fundamental entities to carry out work in organizations (Hackman, 1987; McGrath, 1991; Sproull & Kiesler, 1991). Cohen and Bailey (1997) define a group in an organizational setting as an aggregation of individuals who are linked by interdependent tasks, who perceive themselves and are perceived by others as a distinguished social entity, and who are embedded in a larger social system. Group members can therefore be geographically dispersed, but they must be aware of one another. Distributed groups, sometimes called virtual groups, "are groups of people with a common purpose who carry out interdependent tasks across locations and time, using technology to communicate much more than they use face-to-face meetings" (Cramton, 2001). Distributed groups have become prevalent with the recent blossoming of open productions as vehicles bringing together geographically dispersed members to achieve a common goal (Bakker, 2010). One common characteristic of distributed groups is their reliance on communication technologies, enabling them to transcend spatial and temporal boundaries. Following Saunders and Ahuja (2006), in this paper I focus on temporary, distributed groups, which typically engage in a single task to accomplish their goal, disband after the task is carried out and reassemble to tackle other tasks that belong to the broader project. Such temporary (or project-based) distributed groups are ubiquitous in today's organizations and include arrangements such as task forces (Bigley &

Roberts, 2001; Weick, 1993), movie sets (Bechky, 2006; Perretti & Negro, 2007), project ventures (Schwab & Miner, 2008), and teams of software development (Boh et al., 2007).

Analogous to the definition of organizational learning, group learning is a change in the group's knowledge as a result of a gain in experience (Argote, 2013). Group learning is defined as a "process involving the activities through which individuals create, retain and transfer knowledge through their experience of interacting with one another or with shared artifacts" (Argote, Gruenfeld & Naquin, 2001: 370). For instance, the more group members acquire experience working together the more they generate a share understanding on which member is capable at solving what task (Liang, Moreland & Argote, 1995), or they might create new collective routines for better coordinating their work activities (Zellmer-Bruhn & Gibson, 2006). The key element enabling group learning is thus the accumulation of members' experience of working together, a facet of project-based, open collaboration which the data in my study – namely the history of group members' prior collaborations across multiple tasks – provides effective measurement of.

Repeated collaboration among members of a group has been argued to be a major determinant of organizational learning and to increase group performance (Boh, Slaughter & Espinosa, 2007; Reagans, Argote & Brooks, 2005). As participants "[I]nteract over time, they develop common knowledge about tasks, goals and strategies that facilitate their work, which helps them to manage tasks and member dependencies more effectively" (Boh, Slaughter & Espinosa, 2007: 1320). The development of this common knowledge is eased by the fact that members embedded in close collaborative relationships face a higher motivation to share knowledge with a coworker (Reagans & McEvily, 2003), extending the benefits of their experience beyond their own individual learning.

The explanations put forward for this easing of knowledge-sharing and the concomitant improvement of group performance divide broadly into two factors: (i) cooperative norms and (ii)

reputation. Socialization of individuals within clusters of repeated collaborative ties favors the establishment and internalization of group norms, including cooperation. According to this perspective, knowledge transfer is likely to occur because of the development of shared norms and increased trust gained by working together (Granovetter, 1992; Uzzi, 1996). Reputation is also correlated with repeated collaboration because social control is high, meaning that the risk of retaliation in response to anti-cooperative behavior increases as cohesion of collaborative ties increases (Coleman, 1990).

Besides knowledge-sharing, knowledge storage is also determinant in unveiling the relationship between repeated collaboration and group performance. Knowledge storage involves the use of the group's transactive memory system (TMS), which refers to "shared system that individuals in groups and organizations develop to collectively encode, store, and retrieve information or knowledge in different domains" (Argote & Ren, 2012: 1377). TMS has been shown to facilitate group learning as well as group performance in many different industries such as engineering, industrial design and software development (Liang et al., 1995; Lewis, 2004; Faraj & Sproull, 2000). The mechanisms through which transactive memory improves performance have been thoroughly examined in the literature. Argote and Ren (2012) identify the following steps in showing how TMS boost competitive advantage: (i) TMS enables the formation of meta-knowledge directories of who knows what in the group. These directories provide members access to greater pool of knowledge than they possess on their own; (ii) as a consequence, when these directories of meta-knowledge are shared among group members, TMS induce groups to specialize, so that members who develop a specific experience in a particular field are the ones who become responsible for storing relevant knowledge and carrying out tasks in that field; (iii) by carrying out tasks in a certain field a member increases his or her legitimacy and trustworthiness, which other members are more likely to rely on. (iv) group members channel new knowledge in the same field to the person they perceive as expert, which further reinforces individual specialization (v) finally directories containing meta-knowledge of who knows what crystallize and

members are facilitated in knowing whom to address for information and whom to trust for carrying out tasks in specific field, a process that eventually improves group coordination and performance.

Consistent with this view, existing research has broadly suggested that groups improve their performance by learning from the experience of their members. However the relationship between learning and group performance is not straightforward (Druskat & Kayes, 2000). That is, learning does not necessarily generate performance improvements, and, in some circumstances, may even result in performance decrements, as was found by Bunderson and Sutcliffe (2003) and Katz (1982). Bunderson and Sutcliffe (2003) demonstrated that overemphasizing a team learning orientation in the near term can be detrimental for team performance, although a more balanced learning orientation leads to the formation of a more efficient adaptive behavior that in turn is beneficial for performance. In a study of 50 R&D project teams Katz (1982) found that team performance improved over the first two years that members worked together, remained high until around their fourth year, and then declined. Even if these teams had become increasingly competent in working together over a longer term, their rate of improvement would have decreased over time. Both arguments are in line with a general tendency that Behavioral Theory of the Firm scholars called *myopic learning* (Levinthal & March, 1993). According to this view, learning becomes inefficient or dysfunctional under certain circumstances. For instance, excessive repeated collaboration may lead group members to overly focus on existing solutions and habitual routines, which in turn discourages a more wide-ranging search for novel solutions to the ever changing nature of problems they face. Thus, competency traps (Levitt & March, 1988) sometimes make group members stick to sub-optimal solutions and are hence detrimental to group performance. Furthermore, shared mental models developed in earlier repeated collaborations may encourage team members to withhold ideas that could alter the status quo and help maintaining conditions under which the exploration of more novel and riskier solutions is discouraged (Klimoski & Mohammed, 1994; Stasser & Titus, 1985; Cohen & Levinthal, 1990).

All these arguments combined suggest a non-monotonic relationship between repeated collaboration and group performance. If there isn't enough collaboration group members can't establish sufficient common ground to allow knowledge-sharing. However, if there is too much repeated collaboration competency traps may bind group members to suboptimal solutions. Therefore I postulate:

Hypothesis 1: The relationship between a group's prior collaborations and group performance will be an inverted-U-shaped with decreasing returns after a certain point.

To shed light on the controversial relationship between repeated collaboration and group performance, researchers must first open the black box of group processes to examine how expertise diversity moderates the effect of repeated collaboration on group performance. In summary, I argue that a higher degree of task-expertise homogeneity among group members would help to establish trust and common ground at lower levels of repeated collaboration. At the other end, a higher degree of task-expertise heterogeneity would provide groups with a longer history of repeated collaboration the necessary level of specialization needed for the formation of transactive memory systems.

The importance of group work grows as tasks become increasingly complex and intertwined and therefore beyond the capacity of single individuals to complete in isolation. However, as tasks grow larger, teamwork also becomes more challenging, because it entails a significant amount of coordination between its participants, particularly when activities are of an inherently more collaborative nature (Espinosa, Slaughter, Kraut, & Herbsleb, 2007; Thompson, 1967; Van de Ven et al., 1976). Part of the challenge in coordinating collaborative work lies in the integration of the task-related knowledge of the various individuals into the activities of the group, a necessary process which

adds further complexity to the overall teamwork effort (Boh, Slaughter & Espinosa, 2007; Crossan et al. 1999). The term expertise diversity refers to "differences in the knowledge and skill domains in which members of a group are specialized as a result of their work experience" (Van der Vegt & Bunderson, 2005: 533). Existing research investigated the relationship between expertise diversity and group performance extensively (Mannix & Neale 2005; van Knippenberg & Schippers, 2007). For example, Bunderson and Sutcliffe (2002) operationalized expertise diversity by measuring how group members develop specialized expertise in a restricted number of domains or spread their efforts over a larger number of domains. In the former case they become narrow specialists, in the latter case they become broad generalists. They showed that team performance is positively related to team functional diversity as it leads to greater degree of information-sharing. Drawing from Bunderson and Sutcliffe (2002), Boh, Slaughter and Espinosa (2007) explored the impact of diversity on group performance among teams of software developers. The authors showed that generalist developers develop a broader experience that allows them to have a greater exposure to a variety of different opinions, problems and solutions. Hence, these members are better able at reconciling diverse experiences and coming up with more creative solutions. For this reason, teams whose developers have a broader spectrum of experience are more effective at integrating diverse information to solve their problems and accomplish their tasks (Rulke & Galaskiewicz 2000).

Overall, past research found mixed evidence regarding the relationship between group diversity and performance. For example, Ancona and Caldwell (1992) discovered that, albeit being associated with greater communication which in turn predicts innovation, the direct effect of diversity in functional assignments on innovation is negative. Kilduff, Angelmar & Mehra (2000) hinted at different temporal dynamics in the relationship between diversity and performance. The authors found that the most successful teams of managers competing in

a business simulation presented greater cognitive diversity in terms of interpretative ambiguity

early in the game but then exhibited more clarity and consensus over time. Furthermore, van der Vegt and Bunderson (2005) found a non-linear relationship between expertise diversity and group learning and performance and explored the moderating role of collective team identification in disentangling possibly mixed results. In teams with low self-identification with the team, expertise diversity is likely to be negatively related to team performance, as team members tend to perceive each other as out-group members and thus delegitimize and distrust each other's knowledge (Tajfel & Turner, 1979).

I build on this body of existing research on the controversial relationships between expertise diversity, group learning and group performance by arguing that the curvilinear effect of group learning on performance is contingent on expertise diversity. Earlier I posited that groups at low levels of repeated collaboration struggle to create the common ground necessary to initiate group learning behaviors. However, now I add to that prediction by arguing that the lower the diversity in task-related expertise among group members is, the more the negative effect of low repeated collaboration is mitigated. In a F/OSS context, software contributors with low experience working together can benefit from sharing knowledge about process routines and technical concepts. A shared background ease the development of common ground for members communication, which in turn helps coordination due to more precise expectations about future scenarios of team work (Espinosa, Slaughter, Kraut, & Herbsleb, 2007). Many studies in this domain have demonstrated that task knowledge sharing and integration is necessary to induce positive outcomes in group work. However, distributed problem-solving work is characterized by numerous incongruences making it hard for members with low experience in working together to make sense of tasks requirements and communication by other members. Absence of a shared background about specific expertise and skills of group members that haven't worked long together long can be detrimental to the establishment of group cognitive processes such as shared mental models (Klimoski & Mohammed, 1994) and the development of group norms (Uzzi, 1996; Bandow, 1997). Consequently, I anticipate that a homogeneous distribution of task-related

expertise can be especially beneficial for group performance when group members have short history of prior collaboration, and especially detrimental when group members have a long history of prior collaboration.

On the other end heterogeneous expertise spread across members of groups with high levels of prior collaboration could also be beneficial. Groups with strong TMS tend to have worked in close proximity and collaboratively over a long period of time. Members of these groups have grown to understand how the other members work, what jobs they are best at, and what their unique abilities are. For instance, research on TMS indicates that meta-knowledge of who knows what in the team helps coordination because members know whom to address when they need information, and also because members develop expectations about whom to leave the task in a specific domain to be tackled. (Brandon & Hollingshead, 2004; Liang et al., 1995; Wegner, 1987). Knowing who knows what in the team has also been demonstrated to be beneficial in large software teams (Faraj & Sproull, 2000) where the process of integrating individually owned knowledge is crucial (Tiwana & McLean, 2005). Further studies have shown that understanding how individual task-knowledge and expertise may contribute to other group members work practices facilitates the development collective mind (Weick & Roberts, 1993), which in turn induces coordination and efficiency. When groups with a high degree of repeated collaboration are instead characterized by homogeneous expertise, above-mentioned aspects of myopic learning are exacerbated by the fact that individuals find it harder to recognize each other's skills. Summarizing these arguments I postulate:

Hypothesis 2a: The degree of expertise diversity among collaborating group members will moderate the non-monotonic relation between repeated collaborations and performance. Heterogeneous groups will exhibit higher performance than homogeneous groups at high number of

prior repeated collaborations.

Hypothesis 2b: Conversely, homogeneous groups will exhibit higher performance than heterogeneous groups at low number of prior repeated collaborations.

3.3 Empirical setting: bug-fixing in Free/Open Source Software projects

Free/Open Source Software is an optimal setting to study the effect of group processes on performance as most projects are developed by groups of organizationally - and geographically - distributed contributors who often work on a voluntary basis (Lee & Cole, 2003). F/OSS contributors collaborate from around the world and use technology-mediated communications to coordinate their actions (Raymond, 1999). They constantly form and disband fluid groups concentrated around the resolution of specific problems, such as bug fixes or modifications to the software source code. Contributors manage their teamwork through a variety of coordination tools, including mailing lists for technical discussions, a bug tracking system for monitoring and fixing bugs, a CVS ("Concurrent Versioning System") code repository for storing a common version of the source code (Ankolekar et al., 2006).

As an example of open productions Free/Open Source Software projects are characterized by decentralized, self-emerging coordination processes. Teams are for the most part self-managing, often without formalized role structures or officially nominated leaders. Individual contributors may play different functions in the project or move between functions as their tenure with a project increases. F/OSS project development relies mainly on a distributed coordination model. More peripheral participants usually start by learning how to fix bugs, provide documentation, and help socializing fellow newcomers, whereas more core contributors are responsible for project management and software development. However, even though core developers take on the role of project leaders, they

don't exercise hierarchical authority. Researchers have widely shown that direct assignment of tasks to contributors is almost absent and the most common form of coordination is self-assignment (Crowston & Scozzi, 2008). This type of open production creates a significant information dilemma for contributors when they need to decide where to apply their effort (Benkler, 2006). The scarcity of traditional resource allocation mechanisms such as markets and hierarchies (Powell, 2003; Adler, 2001), means that contributors must self-coordinate, and problems must compete for their attention. In a typical open production environment a number of shared artifacts, such as CVS and bug repositories, constitute the social foci around which individuals coordinate their activities. Project contributors distribute their effort to various projects and their shared artifacts, and use artifacts for various purposes. The project artifacts and discussions are often publicly visible to all members of the environment and are open to any participant who decides to contribute his or her attention or effort to the project (Dabbish et al., 2012).

The transparency of work practices in open productions make F/OSS projects an optimal setting for my study of expertise diversity and group learning. F/OSS projects provide quasi-natural experiments for the analysis of the emergence of problem-solving processes in the absence of exogenous coordination mechanisms. Consistent with the literature on F/OSS and open innovation I consider bug-fixing – i.e., the sequence of tasks intended to permanently solve software errors or glitches that cause computer programs to behave in unintended ways – as one of the most prominent problem-solving activities in software projects (Crowston & Scozzi, 2008; Ankolekar et al., 2006). Bug-fixing is a crucial activity for F/OSS communities since it heavily influences the quality of the software produced, and is thus consistently used in the literature to assess software quality and project performance (Yu, Ramaswamy & Nair, 2011; Crowston, Annabi, & Howison, 2003). The process is usually initiated by a contributor who encounters a problem (i.e., a bug) when using the software. The contributor then reports the bug to the project bug repository, such as Bugzilla, which is the focus of

analysis in this study. These repositories are shared artifacts that typically provide the tracking infrastructure for describing, triaging and resolving software bugs. Via the bug repository, contributors engage in focused, transparent conversations about the project and coordinate their distributed effort toward the resolution of its bugs.

A Bugzilla bug report is made up of distinct sections which facilitate the decentralized, repeated collaboration of contributors on the resolution of the bug. When contributors decide to allocate their effort to the resolution of a bug they must go through the complete history of prior activities on that bug to learn what has and hasn't been done thus far. The pre-defined fields within the report contain a variety of information on the bug and the sub-tasks undertaken by group members. Many subfields, like for instance operating system, version, component, product, severity and priority are initially filled in by the contributor when he or she submits the report, but can as well be modified by other contributors who decide to engage with the bug. Several other subfields are supposed to be continuously modified, to reflect current information such as the status of the report, the contributors who got assigned to solve the bug, the current resolution status. Bug reports contain also a "cc list" with all emails of contributors who decided, or were asked by someone else, to be kept informed about work on the bug. There are two sections within a bug report devoted to text-based interactions: a description box and a comments section. The former contains a comprehensive description of the effects of the bug and the necessary information for a developer to reproduce the bug. The latter contain any comments posted by any contributor such as a discussion of possible fixes or clarifications about specific features of the problem under investigation. Contributors also provide attachments to bug reports. Attachments are software patches that are intended to fix the defective behavior of the program. Attached patches must be reviewed by a different person who verifies the fix and declares the bug "fixed" and eventually "closed". Like other bug-tracking systems, Bugzilla tracks the complete activity of any contributor around every report submitted to the repository. This sort of activity log

offers the complete history all past modifications to each subfield of the report, including for instance all contributors who were assigned to it, or all modifications to the bug severity level. It also provides information about the type of activities that contributors undertake to fix bugs, and thus allows researchers to build accurate measures of task-expertise diversity among members of a group working on a bug.

Bug repositories are shared artifacts that represent the task around which temporary, distributed groups self-assemble. Bug-fixing is a distributed, collaborative task that constitutes a "microcosm of coordination problems and mechanisms to solve them" (Crowston, 1997: 173), allowing contributors to work interdependently and sequentially. As Eric Raymond describes in his *The Cathedral and the Bazaar,* the contributors experiencing and reporting bugs have different expertise from the those who understand the problems and those who solve them (Raymond, 1999). In fact, bug resolution involves the broadest range of project contributors with different expertise, all participating in the process by the means of repeated interactions over shared artifacts – the bug reports. Thus the transparent acts of engagement underlying bug-fixing yield rich data about the complex problem-solving dynamics of distributed, temporary, task-focused groups, which are themselves embedded in a broader open production. Furthermore, the limited role played by exogenous structures, such as hierarchy or departmentalization, make this setting especially apt for studying the emergence of learning dynamics between interdependent actors in non-hierarchical, non-formalized organizational structures. These reasons provide a compelling rationale for the use of data from bug-tracking systems as the basis for the analysis reported in this study.

3.4 Methods

3.4.1 Data and sample

Apache HTTP server is the most widespread web server on the market, with a share of 47% at the time of writing, followed by Microsoft IIS with the second largest share at 22%. Apache is a software program installed on Internet servers hosting web pages. It responds to requests made by users through their browser clients by providing appropriate content over the web. Early versions of Apache developed in the nineties provided quite simplified features. However, more recently all web servers, including Apache, started integrating new functionalities to incorporate more technically advanced features that were being developed over the World Wide Web. For instance, web servers are now utilized to provide gateways to databases, gaming infrastructure and data storage, or to run enterprise applications for use by organizations, rather than individuals Apache is now made of hundreds of specialized libraries which altogether cover the very diverse array of functionalities that are required from web server programs.

The Apache server developed into an independent project in 1995 led by a group of eight core volunteers who joined forces after discovering they had been independently modifying and improving on NCSA HTTPd – the dominant web server software of the time. After having quickly overtaken NCSA as the most widely adopted webserver solution on the Internet, the Apache server became the base for the creation of a common platform for open source software development called Apache Group. In 1999 the Apache Group was incorporated as a non-profit organization known as the Apache Software Foundation (ASF). Drawing on the experience acquired while collaborating on the Apache server, the goal of the foundation is to organizationally, financially and legally support the development of open source software. Like each of the other 140 projects operating under the umbrella of the ASF, the development and maintenance of the Apache server is now overseen by a Project Management

Committee chaired by its own vice president. All appointees to the committee are selected from among the volunteers who have acquired most significant merit in the project through their contributions to it. Meanwhile the right of contributors to modify the software, within Apache, and within all other projects operating under the ASF, is assigned by the community of developers responsible for the development of that particular project, and is earned by showing commitment and active engagement. Newcomers looking for ways to start contributing to a project[2] are explicitly encouraged on the ASF website to find an issue with the software that stimulates their own interest and to engage with it by exploring the reported bugs, improving existing bug reports, and eventually submitting software patches.

I collected data on the complete set of problem-solving activities recorded in the official bug repository of the Apache server. The dataset I generated spans the first bug report reported for the Apache server (Version 2) in September 2001 and the latest bug report at the time of data collection in March 2013. All software bugs engaged with by contributors during this timeframe are included in my dataset and all actions taken while working on them are considered. The raw data were collected by parsing the web pages of all bug reports included in the dataset with the software, *Bicho,* (Robles, Gonzalez-Barahona, Izquierdo-Cortazar & Herraiz, 2009) in order to reconstruct the sequences of timestamped actions taken by individual contributors to modify them. In the final dataset the lifetime of a software bug starts at discovery and ends if and when it experiences resolution. Multiple spells are recorded for each software bug, with a new spell in the lifetime of a software bug starting every time the general state of the bug changes as the consequence of an action taken by a contributor. To build the final dataset I stepped through the sequence of actions extracted from Apache's bug repository. An action, *r*, by contributor *i* at time *t* in the sequence determines the start of a new spell for the software

2 See http://community.apache.org/gettingStarted/101.html

bug, *j*, on which the action was taken. At that point the sequence of actions taken on bug *j* before time *t* is used to measure the values expressed by bug *j* at time *t* – *i.e.* in its current time spell – for all the variables described hereafter. This process yielded a dataset comprised of 5646 software bug reports around which 2630 individual contributors formed distributed, temporary groups, with a total of 24,333 different actions being taken. However, as I am interested in collaborative problem-solving and distributed group dynamics I kept in the sample only those bugs engaged with by groups of two or more contributors. The result was a final sample size of 3470 bug reports and 2143 contributors. An average of 9.25 actions per contributor and 4.3 actions per bug were recorded.

3.4.2 Measures and variables

The dataset contains the following variables:

Software Bug Resolution The dependent variable is time to software bug resolution. A bug is resolved when an action is taken by a contributor to add a 'resolved' flag to its report. More specifically, a bug experiences resolution when one of the following conditions verifies: that the reported problem cannot be reproduced by contributors or that it is reproducible but is considered to be intended behavior and not an error; that the reported problem or suggestion is valid, but that the implementation of a solution is considered unachievable without undesirable side effects; that the problem has been reported before, irrespective of the current resolution status of the original bug report; that a code change which fixes the reported problem is merged in the code base (i.e., resolution by fixing). A software bug flagged as resolved might be reopened at a later date if, for example, the proposed solution is found not to be viable or if new developments in the code base prompt the resurfacing of an issue previously considered non-reproducible. As a consequence, software bugs in the dataset may experience multiple resolution events. Finally, right censoring exists in the dataset whereby software

bugs remain alive and unresolved at the end of the observation period.

Repeated Collaboration I measured repeated collaboration by Apache contributors working on a software bug at each time spell. To do this I began bv examining the sequence of actions extracted from the Apache server's bug repository. An action, *r*, by contributor *i* at time *t* in the sequence determines the start of a new spell for the software bug, *j*, on which the action was taken. I then looked back through the sequence of actions taken on bug *j* before time *t* and counted the number of prior collaborations on other software bugs among all contributors who acted on bug *j* alongside contributor *i*. Prior collaboration is defined as the number of times both contributor *i* and another contributor have both acted on a software bug other than *j* in the past, hence the count gives the number of repeated collaborations which will have occurred when contributor *i* takes action on bug *j* at time *t*. I limited the count of past collaborations by adding a moving time window of six months prior to time *t*. In other words, I only consider as evidence of a prior collaboration the situation whereby co-contributors to bug *j* previously contributed jointly to another bug, insofar as all of those contributions to the other bug occurred within a maximum of six months of each other. The formula I adopted to count repeated collaborations is the following (see also Figure 3.1 for a visual configuration):

$$\sum_{r=2}^{t-1}\sum_{k=1}^{n}\sum_{m=1}^{p}\left(y_{imr}y_{kmr}y_{kjr}\right)$$

where *y* is a problem-solving action, *i* is the focal contributor, *j* is the focal bug, *m* are bugs on which *i* worked in the past, *k* are other contributors that have collaborated with *i* on *m*, and *r* is just the event count.

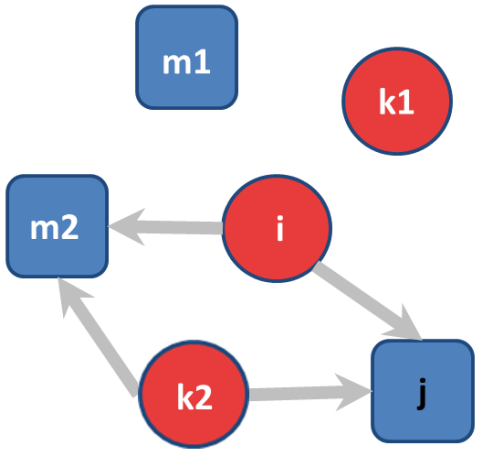


Figure 3.1: Repeated Collaboration visual representation

Task-specific expertise diversity In order to measure diversity in the expertise applied by contributors to the resolution of software bugs I firstly used the Bugzilla classification of the specific actions that contributors can take (see Section 3 for a detailed description). This classification groups actions into eight categories representing different types of action that contributors might take and the expertise needed to perform them. –These categories are as follows: (1) identification actions– these are actions taken to add or correct specific information about the aspect of the Apache server's code base that the software bug is affecting, and which capture knowledge about the direct effect of the issue on the software (e.g., information relating the bug to the component, version and/or hardware it affects); (2) monitoring actions– these are actions that provide an update on the current stage reached in the bug-fixing process started by a specific bug report, and which capture procedural knowledge about the latter (e.g., checking whether a bug should be defined as new, confirmed, assigned, resolved, or closed); (3) descriptive actions – these add or improve on the overall description of the software bug affecting the Apache server and capture knowledge about the internal logic of the issue (e.g., adding comments or summaries); (4) code contribution actions – these report on the work done on software patches and capture the programming skills possessed by contributors attempting to solve an issue

(e.g., submitting or reviewing a submitted software patch); (5) triaging actions – these entail setting and adjusting the prominence of a bug report in the overall bug-fixing workflow for the Apache server and capture knowledge about the latter, as well as the relative importance of a specific issue in the economy of the project (e.g., increasing priority and severity levels of a software bug, or changing its target milestone); (6) dependency identification actions– these identify other software bugs that interact directly with the bug in question and capture knowledge about the full list of issues affecting the code base as well as their potential interdependencies (e.g., identifying the bug dependence tree of a software bug, or other bugs that are blocked by the focal one); (7) specialist identification actions– these identify contributors potentially most interested in, and suited to, solving a specific software bug and capture knowledge about the skill-set of the population of contributors currently active (e.g., officially assigning the software bug to a contributor or putting someone's name on the CC list); (8) resolution decision actions– these record whether one of the conditions required for a software bug to be flagged as solved was met by a software bug (see '*Software bug resolution*' above) and capture knowledge about the direction taken by the evolution of the code base, also in terms of features and intended behavior (e.g., assessing the type of resolution state a software bug might have reached, if any).

Using information on the classification of actions into these categories, I measured diversity in the expertise of contributors collaborating on a software bug by examining the categories of actions they had performed in previous collaborations. I started with the sequence of actions extracted from the Apache server's bug repository. An action, *r*, by contributor *i* at time *t* in the sequence determines the start of a new spell for the software bug, *j*, on which the action was taken. At that point I looked back through the sequence of actions taken on bug *j* before time *t* and determined on which other software bugs other contributors repeating a collaboration with contributor *i* on bug *j* had collaborated. Consistent with my measure of repeated collaboration I limited the count of past collaborations to a

moving time window of six months prior to time *t*. For each of the contributors identified as repeating a collaboration I then constructed an individual expertise profile by counting the number of actions in each category performed on all the software bugs they collaborated on. I finally calculated the Euclidean distance between these profiles to capture the tendency toward expertise diversity expressed by the contributors active on bug *j*. The measure I used to capture the distance among individual expertise profiles is the following (see also Figure 3.2 for a visual configuration):

$$\frac{1}{n}\sum_{k=1}^{n}\sqrt{\sum_{c=1}^{8}\sum_{m=1}^{p}\left(y_{icr}-y_{jcm}\right)^{2}}$$

where *y* is a problem-solving action, *i* is the focal contributor, *j* is the focal bug, *m* are bugs on which *i* worked in the past, *k* are other contributors that have collaborated with *i* on *m*, and *c* is the category of problem-solving action performed.

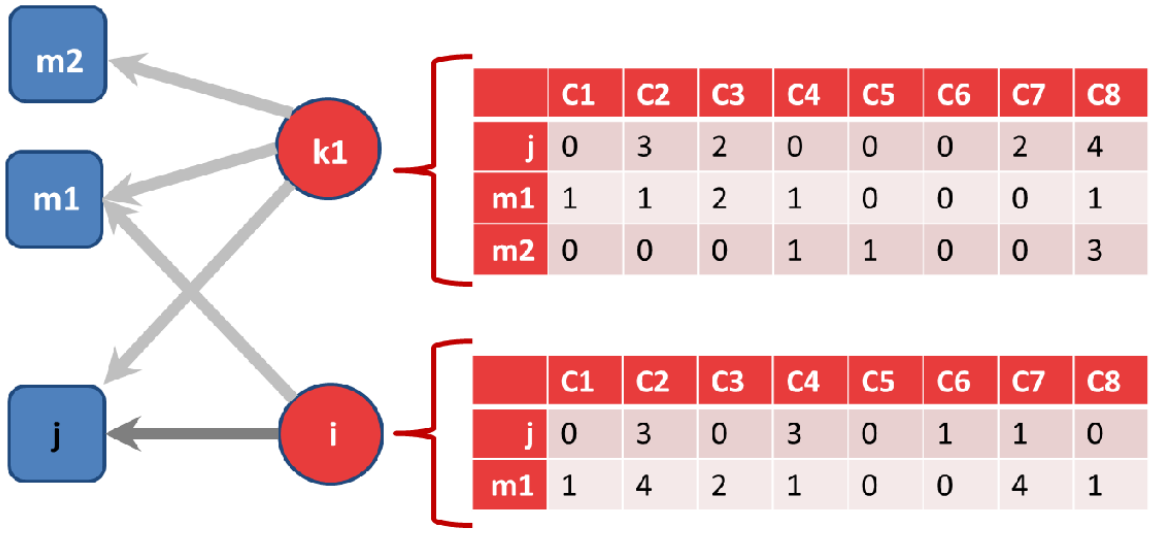


Figure 3.2: Expertise Diversity visual representation

Additional control variables Every time an action, *r*, by contributor *i* is taken on bug *j* I computed a series of measures to control for various individual propensities of contributors and organizational aspects of bug-fixing. I measured the *General tendency to collaborate* manifested by the Apache server's contributors by taking the sum of collaborations entered into by the contributors active on a software bug. At the start of every new time spell for a bug, *j*, the total number of past collaborations on other software bugs involving contributors active on bug *j* was counted – whether these collaborations were repeated on multiple software bugs or not. –. I also calculated the total number of actions already taken by contributors on bug *j* – *Total effort*. In both cases I limited the counts to a moving time window of six months prior to time *t*.

A series of variables was also measured at every new time spell to control for the effect that specific transient characteristics possessed by a software bug at time *t* might have on its resolution. *Software bug severity* was included to control for the propensity to engage with software bugs that are considered more urgent or important by the community of contributors. Contributors can assign a severity level to a software bug on a seven-point scale comprised of the following categories, in order of increasing severity: enhancements, trivial, minor, normal, major, critical, and blocker. Software bugs classified as enhancements are interpreted as requests for new features more than as software failures. The current severity level of a bug was recorded at every new time spell. *Software bug communication* was constructed by counting the number of comments already attached to a bug report, whilst *Software bug CC'ing* counts the number of contributors already added to the CC list. Both measures were included to control for the effect of extra saliency assumed by certain software bugs in the communication flow surrounding bug-fixing. Finally, *Software bug assignee* was constructed by counting the number of contributors already officially assigned to a software bug.

3.4.3 Model specification and estimation

I adopt continuous-time event-history analysis to investigate the time to resolution for any bug reported in the Apache bug repository since its inception. That is, I study the time between a bug's entry into observation (i.e., when a bug is "opened") and a subsequent event. In particular, the event I model is the transition of a bug to the "resolved" state, which happens when a resolution has been performed and there is agreement that the appropriate resolution has been taken. Since bugs can be reopened and resolved again several times, I employ a repeated-event approach in my estimation technique.

Event-history analysis is based on hazard functions defining the risk of observing a specific outcome (e.g., death, failure, resolution) in a time interval *t*, conditioning on the actor having "survived" until time *t*. Hazard functions represent the probability that an entity experiences an event somewhere between *t* and *t* +, divided by the probability that the entity survived beyond time *t*. A useful feature of the event-history approach is the possibility to model temporal variations in the probability of transition to available states, due to the effect of covariates that are multiplied to the hazard rate. I estimate the hazard function using Cox regression (Cox, 1972), a semi-parametric approach used to model event-history data without making assumptions on how is the baseline hazard distributed (Box-Steffensmeier & Jones, 2004; Kacperczyk, 2012). The Cox regression is formalized as:

$$h(t) = q(t) \exp\{\alpha' X(t)\},$$

where $h(t)$ is the hazard rate of a transition to a resolved state at time t, $q(t)$ is a non-specified baseline hazard, $X(t)$ is a the vector of constant or time-varying covariates,, and α' is the coefficient

vector relative to the covariates vector. In particular, following the curvilinear interaction approach of my hypotheses, I specify the vector of covariates as follows:

$$\alpha' X(t) = \alpha_1 C(t) + \alpha_2 C^2(t) + \alpha_3 D(t) + \alpha_4 C * D(t) + \alpha_5 C^2 * D(t) + \alpha' V(t)$$

where *C(t)* is repeated collaboration, *D(t)* is expertise diversity, and *V(t)* is the vector of control variables. A significant benefit for using a Cox regression is that this approach does not assume any specific distribution regarding the effect of time on the baseline hazard line. In particular, the coefficient estimates *α'* reflect shifts in the hazard rate that occur as a consequence of changes in the vector of covariates in *X*, assuming that these changes are proportional within each spell and *q(t)* does not depend on the covariates. The Cox hazard regression is particularly apt for my analysis as preliminary analysis shows no clear parametric distribution relative to the shape of my hazard rate and no clear pattern for time effects on the baseline hazard. An important characteristic of the Cox regression in particular and event-history analysis in general is that it yields reliable estimates even in presence of right-censoring or left-truncation (Tuma & Hannan, 1984). The time period during which right-censored data are recorded terminates before the outcome of interest is observed. For example, a bug could remain open during the whole period of the study, as is the case for some of the bugs in my dataset. Right-censoring techniques address this by allowing units of analysis to contribute to the hazard function only until they are no longer able to contribute, due to the end of observation time. On the other hand there is left-truncation when information on the conditions of the units being studied before the start of the observation time are unavailable (Cox & Oakes, 1984). Since my observation time starts at the very beginning of the project lifetime my sample is not affected by any sample selection bias due to left-truncated observations (Kacperczyk, 2012).

5. Results

Table 3.1 provides descriptive statistics including means, standard deviations, and correlations.

Table 3.1: Descriptives and correlations table

Variable	mean	Sd	1	2	3	4	5	6	7	8
1.Bug comments	5.61	8.86								
2.Bug cc'ing	0.32	0.99	0.45							
3.Bug assignee	0.03	0.21	0.01	-0.01						
4.Bug severity	4.12	1.49	0.02	0.04	-0.05					
5.Bug attachments	0.02	0.14	0.05	0.02	0.04	-0.08				
6.Bug total effort	3.47	3.21	0.38	0.53	0.14	0.07	0.03			
7.Generalized collaboration	69.4	93.35	-0.02	0.04	-0.02	0.08	-0.09	0.18		
8.Repeated collaboration	7.12	2.44	0.03	0.11	0.02	0.05	-0.04	0.20	0.61	
9.Knowledge diversity	54.9	27.9	-0.02	-0.02	-0.03	0.04	-0.04	0.16	0.12	0.19

Since I am interested in collaborative problem-solving practices all bugs that were acted on only by a single contributor were omitted from the sample. During the period between 2003 and 2012, I identified 3,470 bugs with 3,688 instances of bug resolution (recall that a bug can be resolved more than once). Correlations between variables are modest, reducing concern about the possibility of multicollinearity. All variables have been standardized to help the interpretation of the relative magnitude of single parameter estimates (Aiken & West, 1991). Figure 3.3 shows the Kaplan–Meier survival function, the skewness of which indicates that most bugs are likely to be resolved in the first

few days, although there is no clear time dependence.

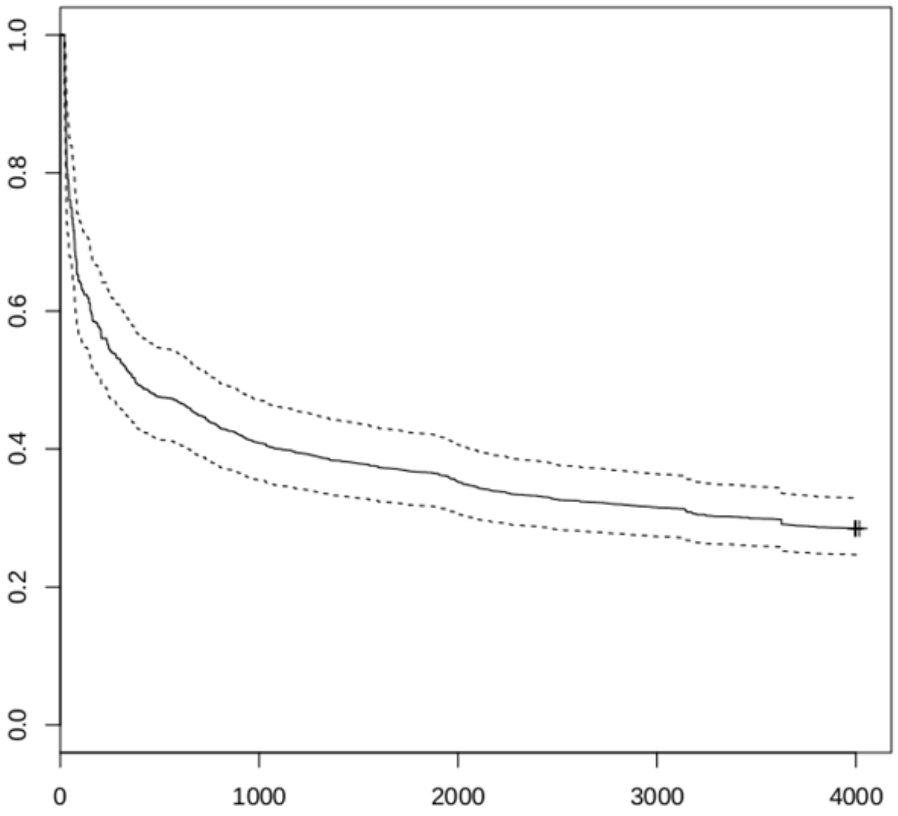


Figure 3.1: Kaplan-Meyer estimator for bug resolution in Apache HTTPD Server, 2001-2013

Table 3.2 shows the results of Cox hazard regressions predicting the time required before a bug is resolved. These models use Huber–White clustered standard errors at the bug level to account for heteroscedasticity.

Table 3.2: Repeated events Cox regression of bug resolution in Apache (robust standard errors in parentheses)

Variable	Model 1	Model 2	Model 3
Bug comments	-0.311** (0.111)	-0.317** (0.109)	-0.311** (0.112)
Bug cc'ing	0.240*** (0.047)	0.239*** (0.048)	0.240*** (0.050)
Bug assignee	0.086 0.050	0.086 (0.051)	0.089 (0.047)
Bug severity	0.117* (0.058)	0.116* (0.058)	0.115 (0.060)
Bug attachments	-2.229*** (0.360)	-2.231*** (0.363)	-2.228*** (0.351)
Bug total effort	0.217** (0.086)	0.224* (0.092)	0.194* (0.084)
Generalized collaboration	-0.831*** (0.250)	-0.814** (0.261)	-0.831** (0.305)
Repeated collaboration		0.744** (0.219)	0.322 (0.286)
Repeated collaboration (squared)		-0.130* (0.065)	-0.151* (0.068)
Expertise diversity			-0.028 (0.034)
Repeated collaboration * Expertise diversity			0.333* (0.158)
Repeated collaboration (squared) * Expertise diversity			0.081* (0.031)
Log likelihood	-28806.32	-27301.43	-27278.75
Wald test (d.f)	1782 (7)	1735 (9)	1626 (12)
N. of observations	21849	21849	21849
N. of events	3831	3831	3831

* $p < 0.05$; *** $p < 0.01$
Robust standard errors are in parentheses

Model 1 is the baseline model with control variables available for the full sample. Model 2 adds the independent variable of interest, *Repeated collaboration*. Model 3 includes the interaction effect of

Expertise diversity with the linear and the quadratic term of *Repeated collaboration*. Estimates from Cox hazard regressions can be exponentiated and interpreted as hazard ratios, similar to odds ratios for logistic regressions. Hazard ratios are the ratio of the hazard rates corresponding to the conditions described by two levels of an explanatory variable, usually having a baseline hazard rate as reference. For example, in my study, a hazard ratio of 2 would mean that an observed bug may be resolved at twice the rate per unit time of the baseline rate, with a unit increase in an explanatory variable.

Model 1 reports parameter estimates for my control variables. A one-standard-deviation increase in the number of comments that a bug raises (*Bug comments*) decreases the hazard that the bug will be resolved by 26 percent [exp(–0.311) – 1], indicating that more complex bugs take longer to be solved. Meanwhile, my alternative measure of complexity, (number of) *Bug attachments*, also significantly affects the hazard of bug resolution. In the case of my *Bug cc'ing* measure, a one-standard-deviation increase in the number of people in the bug cc list, increases the resolution hazard by 27 percent [exp(0.240) – 1], confirming my expectations that more visible bugs have a higher chance of being solved sooner because they attract more attention. Similarly, consistent with my expectations, more severe bugs and tend to be solved faster (i.e., to increase on average the hazard of resolution), showing the effectiveness of formal scheduling attempts. I included two control variables that take into account the baseline dynamics of distributed collaboration. The positive and significant effect of *Bug total effort* indicates that the supply of problem-solving attempts is positively related to the hazard of bug resolution. A single standard deviation increase in total effort increases the baseline rate by 24 percent [exp(0.217) – 1]. On the other hand, the negative and significant effect of *Generalized collaboration* indicates that activity by contributors whose attention is spread across multiple bugs decreases the hazard of resolving a focal bug.

Models 2 and 3 test the substantive validity of my theoretical hypotheses. In Model 2 I included both linear and quadratic terms for *Repeated collaboration* to capture the degree of group learning when individuals repeatedly work together on several tasks. Hypothesis 1 captures the controversial relationship between repeated collaboration and group performance. I proposed that an intermediate level of repeated collaboration is most advantageous for performance, as the benefits of increased group learning are offset in the long run by a myopic reliance on established routines. The linear term for *Repeated collaboration* is positive and significant, whereas the quadratic term is negative and significant, showing an inverted U-shaped relationship between repeated collaboration and the hazard of bug resolution, as I postulated in Hypothesis 1.

Hypotheses 2a and 2b are tested in Model 3. These posit that expertise diversity has a moderating effect on repeated collaboration, such that homogeneous groups outperform heterogeneous groups at low levels of repeated collaboration and that heterogeneous groups outperform homogeneous groups at high levels of repeated collaboration. While the fact that the curvilinear interaction term (*Repeated collaboration (squared) * Expertise diversity*) is statistically significant just tells us that there is a moderating effect of expertise diversity, I need to plot the function to assess precisely the size and the marginal effect of the interaction at different levels of the moderating variable (Dawson, 2014). This is usually done by calculating predicted values of the dependent variable (in my case the hazard rate) under different conditions of the independent variable and moderator (in my case high and low values of repeated collaboration, and high and low values of expertise diversity) and showing the predicted relationship between independent and dependent variable at these different levels of the moderator. Figure 3.4 graphically displays this curvilinear interaction.

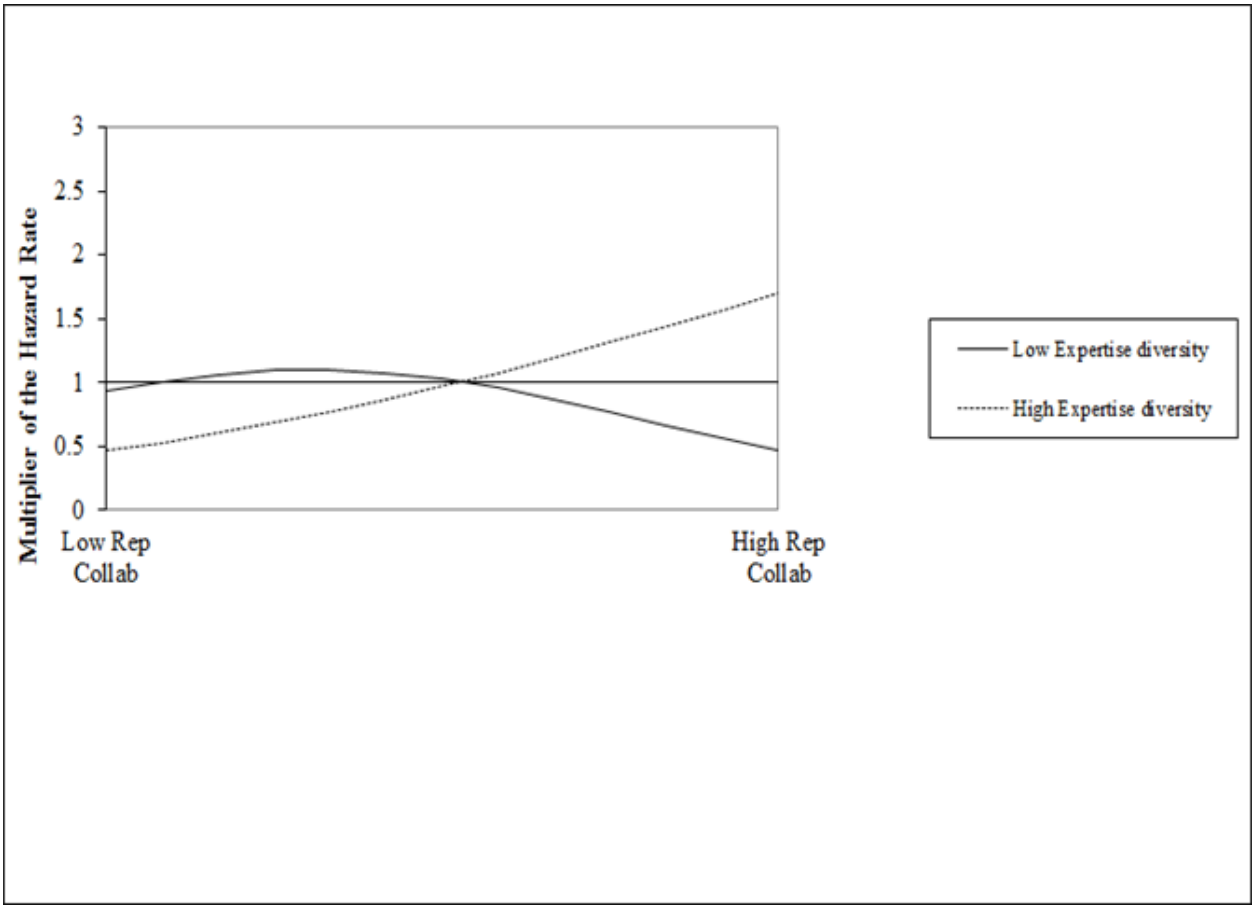


Figure 3.2: The relationship between repeated collaboration, expertise diversity and bug resolution in Apache HTTPD Server, 2001-2013

On the Y axis I plotted the multiplier of the hazard rate (see also Hansen, 1999, and Phillips, 2004 for a similar approach). Analogous to odds ratios, a value above 1 increases the hazard that a bug incurs a resolution event whereas a value below 1 decreases the hazard of resolution. The plot in Figure 4 shows that, ceteris paribus, the hazard of resolution by groups of individuals with heterogeneous expertise working on a bug decreases (i.e., an increase in average time to resolution) if they don't reach a minimum degree of repeated collaboration allowing them to build a sufficient level of common ground and absorptive capacity for effective teamwork (Kraut et al., 2012; Cohen & Levinthal, 1990). However, as the level of repeated collaboration increases, the delay associated with heterogeneous expertise also decreases up to a point where this expertise diversity becomes an advantage, and begins

to increase, rather than decrease, the hazard of resolution (i.e., a decrease average time to resolution). Taking into account the theoretical basis of my hypotheses, it may therefore be argued that the combination of high specialization and a high level of group familiarity helps the development of transactive memory systems, which in turn affect coordination and performance (Liang et al., 1995; Brandon & Hollingshead, 2004). On the other hand, for groups with homogeneous expertise an increase in prior collaboration sees a significant decrease in the hazard of bug resolution beyond a certain threshold. This argument is in line with theories of creative abrasion and myopic learning which identify excessive longevity and lack of specialization as possibly detrimental to the search of novel and better solutions (Katz, 1982; Skilton & Dooley, 2010). Furthermore, by directly comparing the two lines related to low vs high expertise diversity, I can easily assess which group performs better under which condition. As postulated, heterogeneous groups do better than homogeneous group the more they repeatedly work together, whilst homogeneous groups do better than heterogeneous groups the less they do so. This result thus fully supports hypotheses 2a and 2b.

6. Discussion and Conclusions

The results of this research have important implications for the way scholars think about the benefits and challenges of repeated collaboration in distributed, open productions. Specifically, these results further highlight the need to move beyond the simple experience-learning-performance model in order to think in more complex ways about how and under what conditions repeated collaboration in groups might promote or inhibit learning and performance. This study suggests that in order to understand whether a given level of prior collaboration in a group has a positive or negative implication for group performance researchers need to consider the composition of expertise that exists within the group and, more specifically, the extent to which groups are more or less homogeneous in terms of

task-related expertise. In this sample of distributed, temporary problem-solving groups expertise diversity has a significant moderating effect on the relationship between repeated collaboration and group performance: a high level of repeated collaboration could be associated with either a high or low levels of performance, depending on whether and to what extent group members possess heterogeneous knowledge and expertise. This finding extends previous research on the conditions under which prior collaboration might facilitate or hinder group learning and performance by pointing to the significant role of knowledge distribution factors. Furthermore, the results of this research clearly suggest that this moderated relationship between prior collaboration and performance is non-monotonic. Specifically, I find that under conditions of low expertise diversity, the relationship between repeated collaboration and performance is an inverted U-shaped, whereas under conditions of high diversity, the relationship between collaboration and performance follows an almost linear pattern, with the benefits of heterogeneity monotonically increasing with prior collaborations. These results suggest that theories and models of the performance implications of expertise diversity in groups must move beyond linear assumptions in order to accommodate non-monotonic effects. The findings presented here contribute important empirical evidence to support the claim that, under the right conditions, expertise diversity can be a key activator of distributed group learning and thereby promote overall group effectiveness (Van der Vegt & Bunderson, 2005).

This chapter contributes to three main areas of interest. Firstly, this work extends theory on organizational learning by investigating contingent factors that unveil the way in which performance-based learning is achieved in open productions. Support for the moderating effect of expertise diversity highlights the importance of a contingent learning approach, in contrast to prior research that has instead advanced arguments about unconditional and linear effects of experience of working together (Boh et al., 2007). Existing studies (e.g., Reagans et al., 2005) have theorized about the advantages of specialized task-related expertise, which aids the development of transactive memory (theorizing the

role of expertise diversity), and about the value of common ground for coordinating effectively across functions and for sharing knowledge (theory regarding repeated collaboration). However, existing research has often confounded empirically those two concepts, as they have advantages and disadvantages that offset each other. Disentangling the effects of repeated collaboration to those of expertise diversity is important from both a theoretical and an empirical standpoint. By specifying further the interaction effect between these two constructs we are able to advance our understanding about the kind of conditions under which it is preferable to have a diverse versus homogeneous work group. Second, my study has important value for reporting theoretical arguments and empirical findings that advance our knowledge on open innovation and open productions and their distinctive characteristics. This chapter suggests and tests hypotheses that shed light on how teams coordinate and integrate the diverse expertise of their members to solve problems in absence of formal contracts and hierarchies, a central issue in the emerging literature on open production and project-based organizations (Sorenson & Waguespack, 2006; Perretti & Negro, 2007; Bakker, 2010). Finally, this chapter has important value for managers concerned with issues of team staffing and reconfiguration, a topic that has garnered a fair amount of scholarly attention both from academics and in particular practitioners (Reagans, Zuckerman & McEvily, 2004; Groysberg & Lee, 2009). In several contexts, groups with fluid boundaries have become the rule rather than the exception (Huckman et al., 2009). In settings such as creative industries, software development, and consulting, fluid groups are put together to work on a certain task, carry out a certain project and then disbanded. Usually group participants are then reassigned to a new group to work on a new project. My results advocate a more nuanced approach for project managers who have to deal with team staffing issues, a view in which they explicitly consider both experience of working together and degree of knowledge homogeneity amongst team members. For instance, they should be aware of the crucial role of shared homogeneous knowledge and common ground for groups that have low experience working together. At the other

end, they should also be aware that as groups gain experience of working together, more specialized and heterogeneous expertise is preferable for better coordination and group cognition processes.

My study has two main limitations. Firstly, I examined group learning and expertise diversity in within a series of distributed, fluid groups that all belong to the same organization. A research design at the intra-organizational groups level of analysis allows to rule out explanations rooted at the organizational level, although it dampens the generalizability of results beyond that single organizations. Future analysis should confirm results in different organizations and different types of groups. Secondly, when I postulate the effect of expertise diversity at different levels of repeated collaboration I neglect the fact that organizational members learn from each other while working together, so their set of expertise is likely to change over time. Whether groups tend to become more homogeneous or more heterogeneous over time as a consequence of repeated collaboration is an empirical question that I leave open to future studies.

References

Adler, P. S. (2001). Market, hierarchy, and trust: the knowledge economy and the future of capitalism. *Organization science*, *12*(2), 215-234.

Aiken, L. S., & West, S. G. (1991). *Multiple regression: Testing and interpreting interactions*. Sage.

Ancona, D. G., & Caldwell, D. F. (1992). Bridging the boundary: External activity and performance in organizational teams. *Administrative science quarterly*, *37*(4).

Ankolekar, A., Sycara, K., Herbsleb, J., Kraut, R., & Welty, C. (2006, May). Supporting online problem-solving communities with the semantic web. In Proceedings of the 15th international conference on World Wide Web (pp. 575-584). ACM.

Argote, L. (2013). *Organizational learning: Creating, retaining and transferring knowledge*. Springer.

Argote, L., Gruenfeld, D., & Naquin, C. (2001). Group learning in organizations. *Groups at work: Theory and research*, 369-411.

Bakker, R. M. (2010). Taking stock of temporary organizational forms: A systematic review and research agenda. *International Journal of Management Reviews*, *12*(4), 466-486.

Baldwin, C., & von Hippel, E. (2011). Modeling a paradigm shift: From producer innovation to user and open collaborative innovation. *Organization Science*, *22*(6), 1399-1417.

Bandow, D. (1997, April). Geographically distributed work groups and IT: A case study of working relationships and IS professionals. In *Proceedings of the 1997 ACM SIGCPR conference on Computer personnel research* (pp. 87-92). ACM.

Barnard, C. I. 1938. *The Functions of the Executive.* Cambridge: Harvard University Press.

Bechky, B. A. (2006). Gaffers, gofers, and grips: Role-based coordination in temporary organizations. *Organization Science, 17*(1), 3-21.

Benkler, Y. (2006). *The wealth of networks: How social production transforms markets and freedom*. Yale University Press.

Bigley, G. A., & Roberts, K. H. (2001). The incident command system: High-reliability organizing for complex and volatile task environments. *Academy of Management Journal*, *44*(6), 1281-1299.

Boh, W., Slaughter, S. A., & Espinosa, J. A. (2007). Learning from experience in software development: A multilevel analysis. *Management Science*, *53*(8), 1315-1331.

Box-Steffensmeier, J. M., & Jones, B. S. (2004). *Event history modeling: A guide for social scientists*. Cambridge University Press.

Brandon, D. P., & Hollingshead, A. B. (2004). Transactive memory systems in organizations: Matching tasks, expertise, and people. *organization science*, *15*(6), 633-644.

Brown, S. L., & Eisenhardt, K. M. (1995). Product development: past research, present findings, and future directions. *Academy of management review*, *20*(2), 343-378.

Bunderson, J. S., & Boumgarden, P. (2010). Structure and learning in self-managed teams: Why "bureaucratic" teams can be better learners. *Organization Science*, *21*(3), 609-624.

Bunderson, J. S., & Sutcliffe, K. M. (2002). Comparing alternative conceptualizations of functional diversity in management teams: Process and performance effects. *Academy of management journal*, *45*(5), 875-893.

Bunderson, J. S., & Sutcliffe, K. M. (2003). Management team learning orientation and business unit performance. *Journal of Applied Psychology*, *88*(3), 552.

Casciaro, T. (2013). The pursuit of positive affect in task advice networks: Effects on individual performance. *Academy of Management Proceedings*. Working paper.

Cohen, S. G., & Bailey, D. E. (1997). What makes teams work: Group effectiveness research from the shop floor to the executive suite. *Journal of management*, *23*(3), 239-290.

Cohen, W. M., & Levinthal, D. A. (1990). Absorptive capacity: a new perspective on learning and innovation. *Administrative science quarterly*, *35*(1).

Coleman, J. S., & Coleman, J. S. (1994). *Foundations of social theory*. Harvard University Press.

Cramton, C. D. (2001). The mutual knowledge problem and its consequences for dispersed collaboration. *Organization science*, *12*(3), 346-371.

Crossan, M. M., Lane, H. W., & White, R. E. (1999). An organizational learning framework: from intuition to institution. *Academy of management review*, *24*(3), 522-537.

Crowston, K. (1997). A coordination theory approach to organizational process design. *Organization*

Science, *8*(2), 157-175.

Crowston, K., & Howison, J. (2005). The social structure of free and open source software development. *First Monday*, *10*(2).

Crowston, K., & Scozzi, B. (2008). Bug fixing practices within free/libre open source software development teams. *Journal of Database Management (JDM)*, *19*(2), 1-30.

Crowston, K., Wei, K., Howison, J., & Wiggins, A. (2012). Free/Libre open-source software development: What we know and what we do not know. *ACM Computing Surveys (CSUR)*, *44*(2), 7.

Dabbish, L., Stuart, C., Tsay, J., & Herbsleb, J. (2012, February). Social coding in GitHub: transparency and collaboration in an open software repository. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work* (pp. 1277-1286). ACM.

Dawson, J. F. (2013). Moderation in Management Research: What, Why, When, and How. *Journal of Business and Psychology*, 1-19.

Druskat, V. U., & Kayes, D. C. (2000). Learning versus performance in short-term project teams. *Small group research*, *31*(3), 328-353.

Edmondson, A. C., & Nembhard, I. M. (2009). Product development and learning in project teams: the challenges are the benefits*. *Journal of Product Innovation Management*, *26*(2), 123-138.

Espinosa, J. A., Slaughter, S. A., Kraut, R. E., & Herbsleb, J. D. (2007). Team knowledge and coordination in geographically distributed software development. *Journal of Management Information Systems*, *24*(1), 135-169.

Faraj, S., & Sproull, L. (2000). Coordinating expertise in software development teams. *Management science*, *46*(12), 1554-1568.

Faulkner, R. R. (1983). *Music on demand*. Transaction Publishers.

Granovetter, M. (1992). Economic institutions as social constructions: a framework for analysis. *Acta sociologica*, *35*(1), 3-11.

Groysberg, B., & Lee, L. E. (2009). Hiring stars and their colleagues: Exploration and exploitation in professional service firms. *Organization Science*, *20*(4), 740-758.

Hackman, J. R. (1987). The design of work teams. *Ariel*, *129*, 32-197.

Harrison, D. A., Mohammed, S., McGrath, J. E., Florey, A. T., & Vanderstoep, S. W. (2003). Time matters in team performance: Effects of member familiarity, entrainment, and task discontinuity on speed and quality. *Personnel Psychology*, *56*(3), 633-669.

Hinds, P. J., Carley, K. M., Krackhardt, D., & Wholey, D. (2000). Choosing work group members: Balancing similarity, competence, and familiarity. *Organizational behavior and human decision processes*, *81*(2), 226-251.

Huckman, R. S., Staats, B. R., & Upton, D. M. (2009). Team familiarity, role experience, and performance: Evidence from Indian software services. *Management science*, *55*(1), 85-100.

Kacperczyk, A. J. (2012). Opportunity structures in established firms entrepreneurship versus intrapreneurship in mutual funds. *Administrative Science Quarterly*, *57*(3), 484-521.

Kane, A. A., Argote, L., & Levine, J. M. (2005). Knowledge transfer between groups via personnel rotation: Effects of social identity and knowledge quality. *Organizational Behavior and Human Decision Processes*, *96*(1), 56-71.

Katz, R. (1982). The effects of group longevity on project communication and performance. *Administrative Science Quarterly*, *27*(1).

Kilduff, M., Angelmar, R., & Mehra, A. (2000). Top management-team diversity and firm performance: Examining the role of cognitions. Organization Science, 11(1), 21-34.

Klimoski, R., & Mohammed, S. (1994). Team mental model: Construct or metaphor?. *Journal of management*, *20*(2), 403-437.

Kraut, R. E., Fussell, S. R., Brennan, S. E., & Siegel, J. (2002). Understanding effects of proximity on collaboration: Implications for technologies to support remote collaborative work. *Distributed work*, 137-162.

Lakhani, K. R., Lifshitz-Assaf, H., & Tushman, M. (2013). Open innovation and organizational boundaries: Task decomposition, knowledge distribution, and the locus of innovation. *Handbook of economic organization: Integrating economic and organizational theory*, 355-382.

Lee, G. K., & Cole, R. E. (2003). From a firm-based to a community-based model of knowledge creation: The case of the Linux kernel development. *Organization science*, *14*(6), 633-649.

Levinthal, D. A., & March, J. G. (1993). The myopia of learning. *Strategic management journal*, *14*(S2), 95-112.

Levitt, B., & March, J. G. (1988). Organizational learning. *Annual review of sociology*, *14*(1), 319-338.

Lewis, K. (2004). Knowledge and performance in knowledge-worker teams: A longitudinal study of transactive memory systems. *Management science*, *50*(11), 1519-1533.

Liang, D. W., Moreland, R., & Argote, L. (1995). Group versus individual training and group performance: The mediating role of transactive memory. *Personality and Social Psychology Bulletin*, *21*(4), 384-393.

Mannix, E., & Neale, M. A. (2005). What differences make a difference? The promise and reality of diverse teams in organizations. *Psychological science in the public interest*, *6*(2), 31-55.

McGrath, J. E. (1991). Time, interaction, and performance (TIP) A Theory of Groups. *Small group research*, *22*(2), 147-174.

Mintzberg, H. 1979. *The structure of organizations*. Englewood Cliffs, N.J.: Prentice-Hall.

Perretti, F., & Negro, G. (2006). Filling empty seats: How status and organizational hierarchies affect exploration versus exploitation in team design. *Academy of Management Journal*, *49*(4), 759-777.

Perretti, F., & Negro, G. (2007). Mixing genres and matching people: a study in innovation and team composition in Hollywood. *Journal of Organizational Behavior*, *28*(5), 563-586.

Powell, W. (2003). Neither market nor hierarchy. *The sociology of organizations: classic, contemporary, and critical readings*, *315*, 104-117.

Raymond, E. (1999). The cathedral and the bazaar. *Knowledge, Technology & Policy*, *12*(3), 23-49.

Reagans, R., & McEvily, B. (2003). Network structure and knowledge transfer: The effects of cohesion and range. *Administrative science quarterly*, *48*(2), 240-267.

Reagans, R., Argote, L., & Brooks, D. (2005). Individual experience and experience working together: Predicting learning rates from knowing who knows what and knowing how to work together. *Management science*, *51*(6), 869-881.

Robles, G., Gonzalez-Barahona, J. M., Izquierdo-Cortazar, D., & Herraiz, I. (2009). Tools for the study of the usual data sources found in libre software projects. *International Journal of Open Source Software and Processes (IJOSSP)*, *1*(1), 24-45.

Rulke, D. L., & Galaskiewicz, J. (2000). Distribution of knowledge, group network structure, and group performance. *Management Science*, *46*(5), 612-625.

Saunders, C. S., & Ahuja, M. K. (2006). Are all distributed teams the same? Differentiating between temporary and ongoing distributed teams. *Small Group Research*, *37*(6), 662-700.

Schwab, A., & Miner, A. S. (2008). Learning in hybrid-project systems: The effects of project performance on repeated collaboration. *Academy of Management Journal*, *51*(6), 1117-1149.

Skilton, P. F., & Dooley, K. J. (2010). The effects of repeat collaboration on creative abrasion. *Academy of Management Review*, *35*(1), 118-134.

Sorenson, O., & Waguespack, D. M. (2006). Social structure and exchange: Self-confirming dynamics in Hollywood. *Administrative Science Quarterly*, *51*(4), 560-589.

Sproull, L., & Kiesler, S. (1992). *Connections: New ways of working in the networked organization*. MIT press.

Stasser, G., & Titus, W. (1985). Pooling of unshared information in group decision making: Biased information sampling during discussion. *Journal of personality and social psychology*, *48*(6), 1467.

Stuart, H. C., Dabbish, L., Kiesler, S., Kinnaird, P., & Kang, R. (2012, February). Social transparency in networked information exchange: a theoretical framework. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work* (pp. 451-460). ACM.

Tajfel, H., & Turner, J. C. (1979). An integrative theory of intergroup conflict. *The social psychology of intergroup relations*, *33*, 47.

Thompson, J. D. 1967. *Organizations in action*. New York: McGraw-Hill.

Tuma, N. B., & Hannan, M.T. (1984). *Social dynamics models and methods*. Elsevier.

Uzzi, B. (1996). The sources and consequences of embeddedness for the economic performance of organizations: The network effect. *American sociological review*, 674-698.

Uzzi, B., & Lancaster, R. (2003). Relational embeddedness and learning: The case of bank loan managers and their clients. *Management science*, *49*(4), 383-399.

Van de Ven, A. H., Delbecq, A. L., & Koenig Jr, R. (1976). Determinants of coordination modes within organizations. *American sociological review*, 322-338.

Van der Vegt, G. S., & Bunderson, J. S. (2005). Learning and performance in multidisciplinary teams: The importance of collective team identification. *Academy of Management Journal*, *48*(3), 532-547.

Van Knippenberg, D., & Schippers, M. C. (2007). Work group diversity. *Annu. Rev. Psychol.*, *58*, 515-541.

von Hippel, E., & von Krogh, G. (2003). Open source software and the "private-collective" innovation model: Issues for organization science. *Organization science*, *14*(2), 209-223.

Von Krogh, G., Spaeth, S., & Lakhani, K. R. (2003). Community, joining, and specialization in open source software innovation: a case study. *Research Policy*, *32*(7), 1217-1241.

Wegner, D. M. (1987). Transactive memory: A contemporary analysis of the group mind. In *Theories of group behavior* (pp. 185-208). Springer New York.

Weick, K. E. (1993). The collapse of sensemaking in organizations: The Mann Gulch disaster. *Administrative science quarterly*, 628-652.

Weick, K. E., & Roberts, K. H. (1993). Collective mind in organizations: Heedful interrelating on flight decks. *Administrative science quarterly*, 357-381.

Yu, L., Ramaswamy, S., & Nail, A. (2011). Using bug reports as a software quality measure. In *Proceedings of the 16th International Conference on Information Quality (ICIQ)* (pp. 277-286).

Zellmer-Bruhn, M., & Gibson, C. (2006). Multinational organization context: Implications for team learning and performance. *Academy of Management Journal*, *49*(3), 501-518.

Chapter 4

IDENTITY CONSTRUCTION AND SUSTAINED PARTICIPATION IN AN OPEN SOURCE SOFTWARE PROJECT

Abstract

The viability of open productions depends critically on their ability to attract and retain voluntary contributors. This study shifts attention from the principal focus of existing research, on ex-ante motivations – intrinsic and extrinsic – for this voluntary participation, to examine the nature of contributions over time and the consequences for sustaining participation. I suggest that participant retention is critically impacted by the social dynamics of open productions through the progressive socialization and legitimation of participants in the course of their work. Individuals construct their identity profiles as specialists or generalists by narrowly focusing or widely dispersing their efforts across knowledge domains. Drawing from the literature on the sociology of categories, I argue that the expertise of specialist contributors is more easily evaluated by fellow participants, aiding integration and fostering sustained participation. However, I propose a moderating mechanism inherent in the collaborative nature of open productions, whereby the beneficial effect of a specialist identity is diminished as contributors engage and re-engage in collaborative practices. This allows repeat collaborators to broaden their identity profile without damaging their standing within the community, on which their continued participation relies. I test my hypotheses using data on problem-solving in the form of source code contributions to the Apache HTTP Server repository. I examine the duration of the participation of 82 individuals who contribute 1425 commits, involving 10757 files over the time period 1996-2013. The results support my arguments. I discuss the broader implications for the literature on sustained participation in open productions and on the sociology of categorization processes.

4.1 Introduction

One of the most crucial success factors for the viability of open productions is their ability to sustain individual participation and retain human capital (Fang & Neufeld, 2009; Duchenaut, 2005). Existing research has shown that open productions may be seen as operating in a condition of fluid participation – i.e., a situation in which participants vary in the amount of time they devote to productive processes and where organizational boundaries are uncertain and changing (Cohen, March & Olsen, 1972). Fluid participation is typical of open productions because only a minority of organizations in the entire ecosystem ever manage to attract and maintain a level of engagement of contributors compatible with their survival (Lomi, Conaldi & Tonellato, 2012; Krishnamurthy, 2002; Scacchi, 2002). This fluidity also manifests itself in the wide variety of motivational factors affecting decisions to join or leave open productions, which transcend the traditional incentives provided by formal contracts (Lerner & Tirole, 2002). This feature raises an important overarching question to be addressed in the study of success factors for open productions: What mechanisms sustain long-term voluntary participation within them?

As notable examples of open productions Free/Open Source Software (F/OSS) projects are characterized by continuous inflows and outflows of – mostly voluntary – participants. Notwithstanding the well-known success stories, most F/OSS projects have struggled to attract and sustain a sufficient level of contributors' participation (Crowston, Annabi & Howison, 2003). Open productions fail to survive without voluntary contributions. Since contributors are often volunteer participants, rather than standard employees, open productions do not rely on traditional employment contracts and incentives to retain their human capital (von Krogh, Spaeth & Lakhani, 2003). Although the crucial question of the dynamics of voluntary contributions to open productions has spurred extensive research into individual participation in F/OSS projects, most studies to date have primarily focused on identifying individuals' motivations for getting involved in a project (Roberts, Hann & Slaughter, 2006). Usually the implicit assumption in this line of

research is that an ex-ante condition exists driving motivation to participate and stay in the project. In the literature most motivations fall into three broad categories: extrinsic motivations, intrinsic motivations and internalized extrinsic motivations (von Krogh et al., 2012; Fang & Neufeld, 2009). Extrinsic motivations are primarily instrumental and represent a situation where efforts are undertaken in order to obtain a separable outcome, whereas intrinsic motivations pertain to actions that are carried out to fulfil a satisfaction inherent in the action itself rather than a payoff resulting from its consequences (Hars & Ou, 2002). Internalized extrinsic motivations arise when individuals internalize external incentives so that they are perceived as induced by self-regulatory behavior rather than exogenous contingencies (Deci and Ryan, 2000). While prior literature has devoted much attention to the way in which contributors in open productions are intrinsically and extrinsically motivated to dedicate their time and effort to the creation of a public good (Lakhani & von Hippel, 2003; Roberts et al., 2006; von Krogh et al., 2003; 2012), little is known about how and why individuals' sustained participation changes over time as a consequence of the actual work practices that individuals undertake. I argue that there is a lack of substantial attention to the nature of the practices that underlie sustained participation in open productions. In particular, by deciding either to specialize in work within a single knowledge domain or to spread their effort across several domains, organizational participants shape their identity as specialist or generalist (Adamic et al., 2010; Anderson, 2012). Specialist members enjoy certain advantages derived from having a more clearly defined identity in the eyes of their audience, whereas generalist members tend to be devalued or ignored because the diversity of their contributions gives rise to an unclear or confusing identity (Hannan, 2010; Leung & Sharkey, 2014). However, generalist members, whose identity is not constrained by clear expectations from their audience, may enjoy greater role flexibility and freedom of action and may therefore choose to stay in the organization longer (Zuckerman et al., 2003; Becker, 1973). I argue that the degree to which an individual's identity emerges from their day-by-day work practices as more focused or generalized has important implications for our understanding of how sustained participation in open productions is achieved.

Thus, the objective of this study is to address the following two research questions: How do work practices in open productions relate to the construction of participants' identities? How do these identity profiles change over time and affect sustainability of participation? While the existing literature focuses on the reasons why individuals contribute to open productions in the first place, these research questions shift attention to the nature of these contributions over time and the consequences that they have for the sustainability of individual participation. In particular, I draw from the relevant literature on open productions (Levine & Prietula, 2014; Baldwin & von Hippel, 2011) and on the social dynamics of categories (Zuckerman, 1999; Zuckerman et al., 2003; Hannan, Polos & Carroll, 2007) to develop a contingency framework that advances our understanding of the effects of assuming a focused (specialized) versus complex (generalist) identity on sustained contributions to a Free/Open Source Software (F/OSS) project.

In the next part of this chapter I follow Zukerman (1999; 2003) in developing theoretical arguments that shed light on the fundamental trade-off underlying the focused versus complex nature of identities. Both specialism and generalism convey advantages and disadvantages that may offset each other according to specific contingent factors. I consider the roles of experience and repeated collaboration as crucial contingent factors that moderate the relationship between identity specialization and sustained participation. In particular I argue that, to the extent that audiences perceive and evaluate peer participants' efforts according to recognized knowledge-based categories, participants who associate themselves with fewer categories are able to attract greater peer' attention. Participants who focus on simple and recognizable identities by working on fewer domains are more easily evaluated in terms of their expertise and their potential contributions to the organizational welfare and are thus more easily accepted as legitimate members. On the other hand, participants who attempt to develop a more complex identity by making contributions which span many different categories of knowledge risk losing the opportunity to obtain any recognition for their work and thus drop out at higher rates. However, once the contributor has gained enough

experience in one knowledge domain and has repeatedly collaborated with other members her legitimacy as a source of expertise is recognized by her peers. Hence the costs of spanning different categories decrease and generalist members enjoy greater flexibility and fulfillment for being part of an open production with fluid boundaries. The trade-off between the advantages of focused and complex identities is then explained by whether individuals have accumulated enough experience and a sufficient quantity of collaborations to be considered legitimate members of a community.

The paper is organized as follows. After introducing the theoretical arguments and the main hypotheses, I test these by analyzing developers' sustained participation in a F/OSS project, in terms of contributions to the software source code. I use event history models to predict the hazard of a developer dropping out of the project as a function of identity specialization, experience, repeated collaboration and a series of relevant controls. I present the results of my analysis and discuss their implications for the advancement our understanding of participation in open productions and the social dynamics of categorization processes.

4.2 Theory and Hypotheses

4.2.1 Literature review: participation in open productions

Open productions cannot be sustained without the voluntary contributions of groups of individuals who are willing to bear personal costs in terms of time and effort provided in order to help achieve a public good. The question of why individuals voluntarily participate in open production has been a central topic of attention in the relevant literature over the past fifteen years (for a review see von Krogh et al., 2012 and Crowston et al., 2012). In particular, several studies within the existing literature on F/OSS have investigated the broad spectrum of different motivations that individuals may possess when they decide to contribute to the community. For example contributors may participate in order to 'scratch a personal itch' related to specific aspects of the software they care about (Raymond, 1999), to satisfy their own needs (Lakhani & von

Hippel, 2003, Franke & von Hippel, 2003), to feel part of a like-minded community (Ghosh, 1998; Hertel et al., 2003), to gain status within the community (Raymond, 1999).

Much of this motivation-based research sprang up in response to a seminal paper on open source software by economists Josh Lerner and Jean Tirole, who asked a foundational question: "Why do top-notch programmers choose to write code that is released for free? Is this 'gift economy' consistent with the self-interested-economic-agent paradigm?" (Lerner and Tirole, 2001: 821). The two authors advanced an argument grounded in labor economics that put signaling at the center of their theoretical inquiry. They argued that developers' motivation for voluntary participation is derived from the indirect signaling about their capabilities to the software development industry as a whole that such participation allows, with the expected payoff coming in the form of future higher earnings. With the aim of systematizing this heterogeneous body of research Crowston et al. (2012) reviewed early research on this subject to consistently investigate the series of reasons that induce individuals to contribute to F/OSS projects. The studies they reviewed showed that motivations usually belong to one of three families: extrinsic, intrinsic and internalized extrinsic motivations. Crowston and colleague's analysis provides evidence to show that reputation (Lerner & Tirole, 2002) and career development concerns (Hann et al., 2002, Hars & Ou, 2002) are among the most salient extrinsic motivations. Intrinsic motivations show the prevalence of pure fun-based motives such as the enjoyment of sharing or mutual learning opportunities (Ghosh, 1998; Shah, 2006), whereas users' own needs (Markus et al., 2000; Lakhani & von Hippel, 2003) are the among the most frequently cited internalized extrinsic motivations.

Open production scholars have then extended this body of literature by examining what types of motivation sustain individual participation in F/OSS (Shah, 2006; Fang & Neufeld, 2009; Dahlander & O'Mahony, 2011). These studies have consistently found that motivations are not static. For instance, Shah (2006) interviewed contributors from several F/OSS projects who revealed that, notwithstanding the initial push that propels initial participation in the community,

most participants drop out once they achieve what they initially intended. Developers who decide to stay have to move from a merely 'user need' motive to an intrinsic motive, such as passion or fun. Consistent with this finding, Fang and Neufeld (2009) showed that initial motives to contribute are not necessarily related to sustained participation, as developers who contribute long-term are those who are able to create an environment of continuous and fruitful learning.

Building on this latter line of research, this study aims to shed light on the conditions through which an open production system fosters sustained participation which is thus reproduced over time. Like Fang and Neufeld (2009) and Shah (2006) I highlight the evolving nature of contributor activity and its dynamic effect on participation. However, in contrast to these two studies I shift attention from generalized motivations for participation to a much more detailed account of individual work practices that – I argue – constitute the basis for sustained participation by shaping contributors' identities. This approach is similar to that of Duchenaut (2005) and Dahlander and O'Mahony (2011) in considering work activities as the theoretical and empirical lens through which we can understand individual progression in open productions. Duchenaut (2005) investigated socialization processes within the Python project from both a learning and a political perspective, and documented how contributors who are successful in progressing toward the center of the project are strategic in the way they shape the network around them. He showed that to advance in the project it is necessary to be constantly involved in code production and discussions around code production. Understanding the relationships between individuals and code and choosing the right allies in terms of collaboration and discussion helps contributors to become legitimate "socialized" members. Analyzing archival data relating to contributions to the GNOME project, Dahlander and O'Mahony (2011) identified three mechanisms through which developers acquire lateral authority and therefore progress toward the center of the project: technical communication, technical contribution, and coordination work. Furthermore, the authors showed that individual commitment changes as a consequence of career advancement. My work builds on

these efforts to consider work activities as central mechanisms whereby individuals interact with other participants and become socialized in the project. I suggest a novel mechanism affecting the likelihood of sustained participation that has been so far overlooked in the literature, identity construction.

4.2.2 Identity construction in open productions

Research in economic and organizational sociology shows an extensive effort to understand how classification structures and categorization impacts on individual and organizational outcomes (Zuckerman, 1999; Zuckerman et al., 2003; Hsu, 2006; Hannan, 2010). In organizational sociology categories can be thought of as socio-cognitive partitions that cluster together similar social entities, such as individuals and organizations, in markets (Negro & Leung, 2013; Zerubavel, 1997). Hannan and colleagues (2007) have formalized this concept by defining categories as labels that audiences apply to clusters of similar social entities. A label crystallizes if audiences reach a certain level of agreement as to which entity belongs to the category and which doesn't. Categories then drive future perceptions of entities labelled within the same cluster, such as the features the entity possesses or the behavior that is to be expected from its constituents (Leung & Sharkey, 2014).

One key finding in this stream of research is that organizations, actors, or products attempting to span multiple categories defy clear evaluation, and are subsequently discounted or ignored. Zuckerman (1999) first proposed an argument linking category-straddling and unclarity of perception. He showed that stock analysts were confused by firms which spanned recognized industry categories. The reason for this confusion rests on the fact that stock analysts specialize by industry, and so have limited capacity to properly evaluate companies that transcend the boundaries of their focused expertise. Companies that straddle categories are therefore at higher risk of being devalued or completely ignored by stock analysts. This 'categorical imperative' has been

investigated in multiple settings and widely supported by empirical evidence (for a review, see Hannan, 2010).

Zuckerman and colleagues (2003) subsequently advanced this line of research further, investigating the role of categories in labor markets. Building on Faulkner's (1983) insights into the typecasting process among Hollywood composers, the authors elucidated a fundamental trade-off regarding the outcomes of categorization in markets. Faulkner explained typecasting as a process whereby the future opportunities of an actor in the labor market are constrained by the current social attributes of that actor, especially in relation to work experience. By studying career patterns of Hollywood feature-film actors, Zuckerman and colleagues revealed typecasting's double-edged nature: actors who specialize in one film genre have a higher chance of being cast again in the same genre, but a lower chance of being cast in a different genre, although they have a higher chance overall of being hired than non-specialists. The underlying idea here is slightly different from that of the 'categorical imperative'. If the categorical imperative is based on the assumption that specialized audiences don't have the cognitive means to understand generalist entities, typecasting is based on the 'signaling' assumption that when actors don't specialize in any category it is because they don't possess the skills to truly master any of them. To the extent that audiences believe that skills are hard to evaluate and result from category-specific learning investments, they regard experience in one category as evidence that an actor does not have the necessary skills to participate in another category. Generalists – or 'jacks of all trades' –thus face a particularly hard challenge in attempting to demonstrate competence in each of the categories in which they appear and are therefore likely devalued to 'masters of none' (Hsu, 2006) who move across categories as a result of inadequacy. Thus, the typecasting process implies that actors whose skills span categorical boundaries face significant difficulty in gaining recognition for this breadth of ability, because they are easily confused with the unskilled. Building on Zuckerman's ideas of categorical imperative and typecasting, I argue that individuals in open productions face a similar challenge when they start

participating in organizational activities. To the extent that different projects reflect the development of distinct skills and knowledge, individuals start constructing their identity by the simple act of choosing how to allocate their effort across project boundaries. The question of whether individuals will continue participating in the open production thus turns from relying on a purely motivational rationale to instead resting on a more dynamic account of how individuals are progressively socialized into the community through their work activities. Identity construction is a key element that is enacted through work practices that entail sustained participation (Fang & Neufeld, 2009; Lave & Wenger, 1990). Individual participation in an open production requires the construction of a recognizable identity, i.e., a process of establishing who one is, what one can do, and to what extent one becomes legitimized and valued by other participants (Handley et al., 2006). This idea is based on theories of social identity, in particular stressing processes of identity regulation – i.e., an individual's response to peers' incentives such as rewards and promotion acts – and those of identity work – i.e., a deliberate effort to form, maintain, or revise one's perception by the relevant audience in relation to social context and work practices (Alvesson & Willmott, 2002). When audiences – i.e., other participants in the project – perceive a contributor's focused, specialist identity they are better able to evaluate his or her knowledge in terms of current skills and potential contributions to the open production, a process which favors that contributor's socialization. Socialized contributors are more likely to become legitimate members and thus gain access to higher level community resources, such as increased status, restricted areas, and community tools. As newcomers get progressively socialized in the project, their knowledge and access to project assets and resources rise, which in turn encourages even greater participation in the future. Therefore I postulate:

H1: The expected time duration over which a contributor's participation is sustained (sustained participation) in the project is positively related to his or her degree of identity specialization.

The more contributors accumulate work experience in one or more categories the more they are progressively socialized in the organization, due to the fact that in open productions each contribution is thoroughly discussed before being approved. As Zuckerman and colleagues demonstrate (Phillips & Zuckerman, 2001; Zuckerman et al., 2003), once individuals gain recognition in their markets or inside their organizations, higher-status individuals can better sustain deviation from the behavior that is expected by labels attached to their associated category. To the extent that open productions are meritocracy-driven environments where status depends on the volume of one's experience, generalist individuals (i.e., individuals with lower identity specialization) with substantial experience are more likely to be perceived by audiences as highly skilled in many categories rather than unskilled in most. This high recognition by peer members then facilitates progress in the project and is thus positively related to sustained participation. Conversely, audiences are also more likely to evaluate highly experienced contributors with a narrow identity as indicating a limited skill set. Therefore, following Zuckerman's and colleague's logic, I hypothesize:

H2a: Contributor experience moderates the relationship between specialization and the time duration over which participation is sustained in the project (sustained participation). In particular, the positive effect of specialization on sustained participation is higher when experience is lower.

H2b: At a high volume of experience, the expected duration of sustained participation in open productions is higher for generalist contributors than for specialist contributors.

A second and more novel contingency effect comes from the empirical evidence that, far from working in a vacuum, individuals constantly interact with other participants, co-operating in ongoing discussion, evaluation and re-evaluation of work on a project. The result of this iterative interaction process is a reshaping of contributor identities over time. Some researchers have

suggested that contributors move from peripheral participation toward the center of the project in a manner akin to classic apprentice-master learning dynamics (von Krogh et al., 2003; Duchenaut, 2005; Dahlander & O'Mahony, 2011). However, in F/OSS almost all communication is technology-mediated, hence the newcomers lack most of the tangible means for learning from the "master", such as the interaction face-to-face. Nevertheless, mailing lists and collaboration artefacts constitute not simply the end product of efforts in open productions, but also the material infrastructure that F/OSS contributors use to interact with one another. As Duchenaut (2005) described in one of the few attempts to address this issue in the context of open source software "The multiple components of an Open Source project that at first seem to be hard material are in essence text. This distributed network of people and things is constructed through the materialization of language. There is a hybridism of dialogue and code, where the dialogue is directly embedded in the code" (Duchenaut, 2005: 327). When contributors engage in collaborative problem-solving activities in open productions they interact over a shared infrastructure that fosters extensive communication and reciprocal feedback processes. Increased feedback and communication aide identity-construction and legitimation of the expertise of individual contributors to a project, thus encouraging sustained participation without the need to incur the costs associated with having a specialized profile. Conversely, as in the case of highly collaborative individuals, generalist members – i.e., members with low identity specialization – will benefit from the flexibility and freedom of not being attached to any specific label while still being perceived as legitimate members in the community Therefore I posit that:

H3a: Repeated collaboration by a contributor moderates the relationship between specialization and sustained participation in the project. In particular, the positive effect of specialization on the duration of sustained participation is higher when the volume of repeated collaboration is lower.

H3b: At high volume of repeated collaboration, the expected duration of sustained participation in open productions is higher for generalist contributors than for specialist contributors.

4.3 Empirical setting: code development in the Apache HTTP server

I test my hypothesis using data I have collected on code development in the Apache HTTP server, a large and successful F/OSS project. I focus on code development rather than bug-fixing because this setting allows me to better analyze processes of socialization and identity construction, whereas individuals involved in bug-fixing do not necessarily contribute to progression of the project, and their degree of own use value motivation is higher (Crowston & Scozzi, 2008). I have chosen Apache as a case study because it is a relatively complex project with many software sub-modules that constitute separate environments for specialization, interaction and learning (González-Barahona, López & Robles, 2004). This feature, besides Apache's long lifespan, allows me to observe a great deal of variance in developers "career paths", unfolding as a result of different work activities that in turn shape identity construction.

The community of contributors who are in charge of developing the Apache HTTPD Server is completely made by volunteers (Fielding, 1999), an exogenous condition that I exploit to rule out purely extrinsic pay-based motivations[3]. The voluntary nature of the work means contributors cannot plan to dedicate consistently large amounts of time to the project, a fact which necessitates the development of processes emphasizing decentralized decision-making and traceable and transparent communication. Apache developers rely on transparent mailing lists to communicate with one another, and a democratic system for voting to settle disputes. Due to the widely dispersed geographic distribution of group members, coordination is often pursued through participation in mailing list discussions, sort of "virtual meeting rooms" where conversations happen

[3] See also http://www.apache.org/foundation/how-it-works.html#structure

asynchronously and transparently. Some projects additionally use more synchronous messaging, such as internet relay chat (IRC) or instant messengers. High costs language barriers make voice and face-to-face communication rather rare. The asynchronicity of mailing lists communication makes it a much preferable means of communication, as it allows the creation of transparent archives that are better suited for the coordination of voluntary contributors.

According to the Apache Software Foundation guidelines[4], the Apache projects are managed using a collaborative, consensus-based process. Rather than operating within a strictly hierarchical structure, with higher positions associated with increasing power, authority is exerted horizontally as contributors in different functions are granted different rights and access to the various areas of the organization. Advancement in the project is based on meritocracy. In one of the founder's own words: "The more work you have done, the more you are allowed to do" (Fielding, 1999: 43).

This study focuses on Apache users known as contributors. Contributors are users who provide or modify code or documentation to the project. Contributors have extra rights and responsibilities, above and beyond those of ordinary users, in terms of activity on the mailing list for developers, contribution of code, patches, documentation, feedback, and participation in discussions on development issues. Contributors are also called developers. Depending on their contribution developers may progress to the status of committer. As soon as they demonstrate their quality by having contributed for a protracted amount of time (i.e., usually six months) they are nominated for membership as committers and are then subject to a vote for membership by the existing committers. Contributors who become committers sign a Contributor License Agreement, gain access to write to the code repository so they can commit code and patches autonomously. Committers vote whenever a new piece of software is added or modified, and are allowed to access and commit code changes to the code repository which is based on Subversion, an open source Concurrent Versions System (CVS). Even though the process for software development is not

[4] See http://httpd.apache.org/dev/guidelines.html

unique nor formalized, Apache contributors undertake a fairly routinized list of actions when they commit new software code. This series consists of: identifying room for improvement on a number of lines of source code; deciding whether someone will work on it; laying out and discussing a possible improvement; downloading locally a copy of the source code for development and testing; presenting and discussing for review the modifications to the code to the mailing list; and finally, sending the code commit with its relative documentation to the CVS repository. This process may take several attempts before reaching a satisfactory conclusion, depending on the scope of the potential improvement, although it is advised that the whole set of source code lines changed should be included in a single commit. Every commit yields a review file with a description of all modifications to the code, a patch for testing them and a complete log of the actions taken. This summary is then submitted to the developers' mailing list for review. For each commit all of the members in the developer community are accountable for checking the mailing list to make sure all modifications are relevant and pertinent.

4.4 Methods

4.4.1 Data and sample

To test my hypotheses I created a unique longitudinal dataset containing the complete history of code commits to the Apache HTTP Server CVS repository, and later to its successor, Subversion. In open source software development contributors identify, program and submit modifications of the software source code to a centralized repository where other contributors who have access to it have the right to discuss, peer-review and vote on submitted changes. I mined both the CVS and Subversion repositories using the web parser CVSAnalY (Robles, Koch & Gonzalez-Barahona, 2004). This tool retrieves the information about every commit to the repository, and dumps it into a database where it can be conveniently analyzed. To avoid duplication of issues and concentrate on the core development of the software I restricted my sample to changes committed

to the Apache HTTP Server trunk. In software development jargon, trunk refers to the unnamed branch of a file tree under revision control. The trunk represents the base for software development over which a project builds its progression.

I have relied on several sources to gather the data on developers contained in the dataset. The Subversion repository parsed with CVSAnalY provided information on the time stamp of the commit, the username and ID of the committer, and the full pathname of changed files. I used the data collected through this repository to derive the specialism variable of theoretical interest and the two moderating variables, *Cumulated experience* and *Repeated collaboration*. I then used the Ohloh online database – an open hub that gathers extensive data on several F/OSS projects and their contributors – to cross-check available information and to search for additional information on the committers' background for use as control variables, such as activity in other projects and type of known programming languages. Finally, I used the Apache bug repository and mailing lists to get information on the first instance of activity in the project for each of the contributors. Relying only on information on first commits to capture the beginning of participation in a project is unreliable as commits usually follow a time period of adjustment and learning that typically results in communication and bug-fixing activities (von Krogh et al., 2003)

To construct the sample I collected data on all of the commits submitted to the Apache Subversion repository for the time period from 1996 to 2013. During this time period 111 developers were uniquely identified as contributing to the repository with a total of 10757 files changed. Using the information from the complete paths of the changed files, I reconstructed the affiliation of each file to the 23 sub-modules that constitute the Apache HTTP server. To build the final dataset I gathered information using the whole sequence of actions parsed from Apache's code repository. Multiple spells are recorded for each contributor, with a new spell in the lifetime of a contributor starting every time his or her status changes as a result of committing a new set of lines of code to the Subversion repository. A commit action, *r*, by contributor *i* on software module *j* at

time *t* in the sequence determines the start of a new spell for committer *i*, who performed the action. Every time a new action occurs, the sequence of actions performed by contributor *i* before time *t* is taken to code all the variables relative to contributor *i* at time *t*. This process resulted in a dataset containing 1454 commits performed by 111 individual contributors on a total of 10757 files being changed. However, in the final sample I decided to retain only those contributors who had submitted more than one commit. The reasons for this are twofold. Firstly, one-time commits are usually the consequence of an idiosyncratic need on the part of the user which requires them to modify a particular feature in the software source code. These commits are more likely to result from a "user value" type of motivation rather than from any intention to contribute to the public good. Secondly, the longitudinal process of identity construction I seek to shed light on in this study is based specifically on multiple instances of collaborative problem-solving. As such, single instances are not relevant. This procedure resulted in a final sample size of 82 contributors and 1425 commits[5]. An average of 20.2 commits per contributor was recorded. However, in line with the existing literature on open productions, there is a core of contributors whose actions account for the majority of overall activity, with 22% of individuals responsible for 71% of all commits.

4.4.2 Variables and measures

Dependent variable. The goal of this chapter is to examine the conditions under which individual participation is sustained in an open production. I measure contributors' *sustained participation* in the project using individual tenure, that is, the time duration of contributor participation in the project. Existing research has used individual tenure with projects as a measure of participation. Howison and colleagues (2006) reported a highly skewed distribution for participation in F/OSS. In the 120 SourceForge projects that they investigated, the authors found that usually contributors don't last more than a month in a project, independently of the role taken.

[5] Models estimated using the unrestricted dataset yielded similar results.

However, the duration of participation may vary in a substantial way according to the different roles undertaken by contributors. For instance, Robles and Gonzalez-Barahona (2005) reported especially long tenure among Debian package maintainers compared to other contributor roles within that operating system.

At any point in time a contributor is at risk of ending his or her participation. To codify exit from the risk set and distinguish active from inactive developers I calculate the maximum amount of time elapsed between two commits by the same individual, and subtract it from the end of observation time. This gives me a threshold of time for each individual beyond which a contributor is considered inactive. To measure the entry of individuals into the risk set I gathered information from the Apache bug repository and the developers' mailing lists and for each contributor codified the first instance of participation in the project. For contributors whose first action could not be found in either repository I set an entry time equal to the time of first commit minus the average difference between the first commit and the first action, across all contributors.

Since my interest is in sustained participation and the dependent variable technically measures its opposite (i.e., the hazard of leaving the project) I interpret negative and significant covariate parameters as increasing the likelihood of sustained participation. Concomitantly, positive and significant parameters signify, ceteris paribus, a higher chance of contributors leaving the project earlier.

Identity specialization. The main independent variable of theoretical interest is the degree of specialism versus generalism in work practices, which I use as a proxy to measure a focused versus more complex identity. In order to capture this theoretical construct in my data I rely on the highly modular structure of the Apache project. Modularity has been consistently examined in the literature on organizations in general (Simon, 1962; Henderson & Clark, 1990; Baldwin & Clark, 2000) and open productions (Baldwin & Clark, 2006; Langlois & Garzarelli, 2008; Henkel &

Baldwin, 2009). A key element of this line of research can be identified in a principle called the "mirroring hypothesis" (Baldwin, 2008). Henderson and Clark (1990) first related the concept of mirroring to product development groups: "We have assumed that organizations are boundedly rational, and hence that their knowledge and information processing structure come to mirror the internal structure of the product they are developing" (p. 27). Hence, modular systems facilitate the spread of very important organizational features, such as division of labor, within-group specialization and learning. Within module boundaries contributors in open productions invest time and effort to develop specialized skills and knowledge that make such organizational arrangements "nearly decomposable systems" (Simon, 1962), or systems where most interactions take place within modules and few across modules.

In Apache specialization the division of labor is achieved through a highly modular architecture (González-Barahona, López, & Robles, 2004). I identified 23 different sub-modules within the trunk of Apache HTTP Server project[6]. These sub-modules reflect different areas of software development that require specialized programming skills to be addressed. For instance, the module labelled "database" stands for the Apache DBD framework which manages connections to SQL backends efficiently, whereas the module labelled "ssl" is a directory housing code for OpenSSL functionality. I tracked information regarding file pathnames contained in all of the contributors commits over time and identified which modules the files refer to. I started with the sequence of actions extracted from the Apache server's CVS and Subversion repositories. An action, *r*, by contributor *i* at time *t* in the sequence determines the start of a new spell for the contributor *i*,. At that point I looked back through the sequence of actions taken by contributor *i* on module *j* before time *t* and determined on which other modules the contributor had worked in the past. I then constructed an individual expertise profile by counting the number of commits in each module. I finally computed a Herfindal-Hirschmann Index to capture the degree of specialization

[6] See http://svn.apache.org/repos/asf/httpd/httpd/trunk/modules/README for a complete list.

within modules of contributor *i*. The formula I used to capture *identity specialization* is the following:

$$HHI_i^* = \frac{(\sum_{j=1}^{N} s_j^2 - \frac{1}{N})}{1 - 1/N}$$

Where *i* is the contributor, *j* is the module, *N* is the total number of modules in the project and *s* the number of file modified in each of the modules touched in the commit under analysis. The statistic is normalised using the total number of modules identified in the project (N).

Cumulated Experience

In my interaction hypotheses I argue that two contingency factors – *cumulated experience* and *repeated collaboration* – moderate the effect of identity specialization on sustained participation. I capture cumulated experience by counting, at the time of a new commit, the cumulated number of files modified in the past by the same contributor, normalized by a decay function that applies a higher weight to more recent commits. Again I began with the sequence of actions extracted from the Apache server's CVS and Subversion repositories. An action, *r*, by contributor *i* at time *t* in the sequence determines the start of a new spell for the contributor *i*. At that point I looked back through the sequence of actions taken by contributor *i* and counted the number of file modifications. I then weighted the result by a linear decay function. The formula for *cumulated experience* is the following:

$$CE_i = \sum_{r=1}^{T} s_r \frac{r}{T}$$

where $r\{1\ldots t\}$, t is the current time in the sequence , and *s* the number of file modified at each point in time.

Repeated Collaboration

As was the case in the previous chapter, I measure repeated collaboration as evidenced by Apache contributors working together on the same modules at each time spell. Software development requires a high degree of coordination and collaboration as every changelog that is submitted to the development mailing list has to be reviewed and accepted by other participants before any subsequent change can be made. Furthermore, as different modules entail a different set of skills and specialized knowledge, repeated collaboration takes on additional important in guaranteeing the necessary level of cohesion and common ground required for effective work within module boundaries (Baldwin, 2008). To measure repeated collaboration I began, once again, by examining the sequence of actions extracted from the Apache server's repository. An action, *r*, by contributor *i* on module *j* at time *t* in the sequence determines the start of a new spell. I then looked back through the sequence of actions taken on module *j* before time *t* and counted the number of prior collaborations on the same module among all contributors who acted on module *j* alongside contributor *i*. Repeated collaboration is defined as the number of times both contributor *i* and another contributor have both modified files on module *j* in the past, hence the count gives the number of repeated collaborations which will have occurred when contributor *i* takes action on module *j* at time *t*. The formula I adopted to count repeated collaborations is the following:

$$RC_i = \sum_{r=1}^{t-1} \sum_{j=1}^{J} \sum_{l=1}^{L} s_{ijr} \times (s_{ljr} - 1) \times s_{ijt}$$

where *J* is the vector of modules in the project, *L* is the vector of other contributors *l*

Other control variables

Every time an action, *r*, by contributor *i* is taken on module *j* I computed a series of variables to control for various individual propensities of contributors to stay or leave the project as

a result of specific work practices. As in the previous chapter, I measured Apache contributors' *General tendency to collaborate* by taking the sum of collaborations undertaken by contributors, regardless of the module in which they occurred or of whether the collaborations were repeated. Thus, at the start of every new time spell for a contributor, *i*, the total number of past collaborations on whichever module involved contributors active on module *j* was counted – whether these collaborations were repeated on multiple modules or not. For the second control I computed the total *scope of contributions* already performed by individuals in different modules, to capture the overall exposure to the project. This measure is different from *identity specialization*, as the first is an unweighted, raw count of modules touched by the contributor, whereas the second takes into account the overall activity, in terms of volume of file modifications in each module. I also gathered information on each contributor's *external activities* (that is, external to the focal project), to capture a generalized commitment to the open source community, indicating a higher degree of intrinsic motivation. To do this I codified a dummy variable recording whether a contributor was active in any other open source project prior to time *t*. Finally, I measured contributors' level of *technical capabilities* as represented by programming languages primarily used to develop code changes. The assumption is that the greater the number of different programming languages known by the contributor, the higher are his or her technical capabilities, and the higher the chances of sustaining a fruitful and fulfilling participation in the community over time.

4.4.3 Empirical model specification and estimation

As was the case in my analysis of bug resolution in the previous chapter, I adopt continuous-time event-history analysis, this time to examine Apache contributors' sustained participation in the project advancement in terms of source code changes committed to the CVS repository. Event history analysis may be understood as being based on "failure time process", which is made of entities – such as individuals or organizations – that are observed over time, starting at some

beginning point. These entities are said to belong to some state, for instance the individual is alive, the organization is in business. These entities are said to be at risk of transitioning to another state (e.g., the individual dies, the organization goes out of business). at any particular point in time, and such probability is given by the hazard rate that we are modelling. These stochastic transitions to other states are called "events". Therefore, event-history analysis is based on hazard functions defining the risk of observing a specific outcome (e.g., death, failure) in an interval after time t, conditional on the subject having "survived" to time t. Hazard functions represent the probability that an entity experiences an event somewhere between t and t +, divided by the probability that the entity survived up to time t. To be specific, I model the time between a contributor's entry into observation (i.e., when a contributor submits his or her first commit to the CVS) and a "failure" event. In particular, the event I model is the transition of a developer to the "inactive" state, which was computed as discussed in the previous section. After an entity experiences an event, such entity may transition to yet another state or may be dropped from the "risk set", i.e., the array of units that are at risk of experiencing an event. When, at the end of the observation period, an entity has not made any single transition from one state to another – i.e., has not experienced any event – such cases are said to be "right censored". For example, a contributor could remain active during the whole period of the study and could experience failure after the end of my observation time. Right-censoring techniques address this by allowing units of analysis to contribute to the hazard function only until they are no longer able to contribute, due to the end of observation time.

To test my hypotheses I use Cox proportional hazard models (Cox, 1972) that allow me to make use of the continuous data at my disposal and to account for the fact that some contributors are still participating at the end of the observation time (and are thus treated as right-censored). The Cox regression is formalized as:

$$h(t) = q(t) \exp\{\alpha' X(t)\},$$

where *h(t)* is the hazard rate of a transition to a resolved state at time *t*, *q(t)* is a non-specified baseline hazard, *X(t)* is a the vector of constant or time-varying covariates,, and α' is the coefficient vector relative to the covariates. A useful feature of the event-history approach is the possibility to model temporal variations in the probability of transition to available states, due to the effect of covariates that are multiplied to the hazard rate. In particular, to account for the interaction terms in my hypotheses, I specify the vector of covariates as follows:

$$\alpha' X(t) = \alpha_1 C(t) + \alpha_2 D(t) + \alpha_3 E(t) + \alpha_4 C*D(t) + \alpha_5 C*E(t) + \alpha' V(t),$$

where *C(t)* is identity specialism, *D(t)* is cumulated experience, *E(t)* is repeated collaboration and *V(t)* is the vector of control variables. The Cox hazard models are agnostic about the assumptions of the shape of the hazard function, as long as it is constant within each spell. The coefficient estimates α' reflect shifts in the hazard rate that occur as a consequence of changes in the vector of covariates in X, assuming that these changes are proportional within each spell and *q(t)* does not depend on the covariates. Cox hazard regressions have no intercept, since they are subsumed into the baseline hazard. The coefficients can be interpreted as the change in hazard for a one-unit change in the underlying covariate.

4.5 Results

Table 4.1 reports descriptive statistics for the variables included in my models, encompassing means, standard deviations and correlations. The correlations between variables show that, as expected, the scope of contributions (N of modules) is negatively correlated with Identity specialization, though not so highly as to cause concern regarding multicollinearity. There is also a high positive correlation between Experience and Repeated Collaboration showing that individuals with greater experience have a higher chance of collaborating repeatedly with other

individuals. I retained both of these variables in the model to allow the teasing apart the two effects and to identify the marginal effect of prior repeated collaboration over and above the cumulated number of actions. I standardized all variables to reduce correlations between multiplicative terms and to facilitate the interpretation of the relative magnitude of single parameter estimates (Aiken & West, 1991).

Table 4.3: Descriptive Statistics and correlations

Variable	**mean**	**sd**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
1.N of modules	5.9	6.24								
2.Cumulated Experience	117.79	213.36	0.03							
3.Activity in other projects	0.84	0.16	0.28	0.14						
4.Technical Knowledge	10.1	4.28	0.32	0.35	0.31					
5. Generalized Collaboration	80	64.73	0.63	0.07	0.30	0.34				
6.Repeated Collaboration	3.47	3.21	-0.03	0.65	0.09	0.25	0.04			
7.Identity Specialism	0.64	0.38	-0.72	0.25	-0.26	-0.27	-0.42	0.26		
8.Identity Specialization * Cumulated Experience	96.33	215.18	0.16	0.81	0.21	0.43	0.19	0.69	0.03	
9.Identity Specialization * Repeated Collaboration	54.9	27.9	0.24	0.69	0.19	0.36	0.28	0.84	0.06	0.53

I report the baseline survival function using the Kaplan–Meier estimator in Figure 4.1.

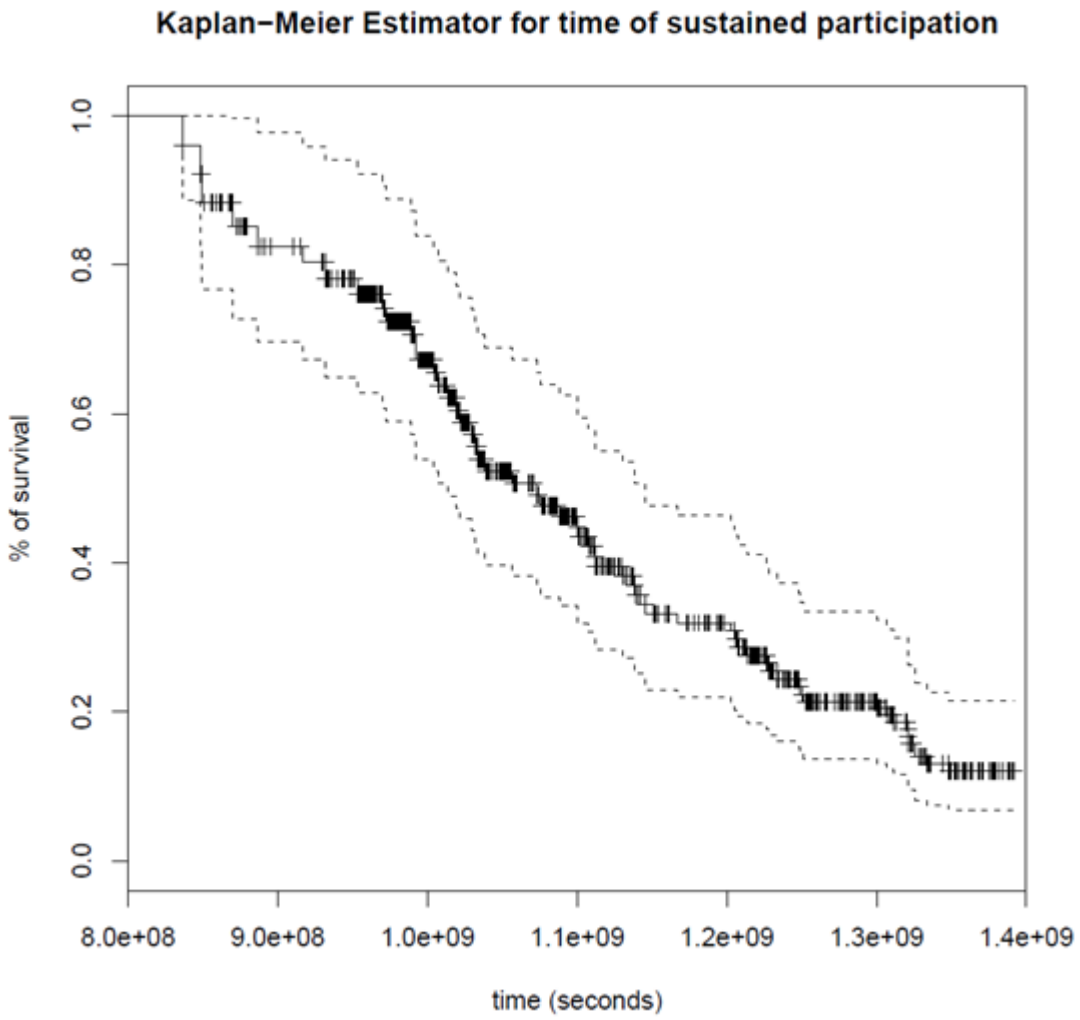


Figure 3: Kaplan-Meyer estimator for time of sustained participation, Apache HTTP Server, 1996-2013

The graph shows the baseline survival times without the inclusion of any covariate. Results of the Cox model are reported in Table 4.2. I used robust Hubert-White sandwich estimators to cluster standard errors around each contributor. Table 4.2 shows the results of Cox regressions predicting the hazard of leaving the project. A negative and significant parameter in the regression would indicate that a unit increase of the relative covariate would decrease the hazard of the

observed individual leaving the project, i.e., increase his or her chances of continuing to participate in the project.

Table 4.4: Repeated events Cox hazard regression of bug resolution in Apache (robust standard errors in parentheses)

Variable	Model 1	Model 2	Model 3	Model 4
Scope of Contributions	1.155*** (0.288)	0.779* (0.317)	0.080 (0.507)	0.263 (0.547)
Cumulated Experience	0.112 (0.269)	0.359 (0.284)	0.991* (0.489)	2.109* (0.956)
Activity in other projects	-0.206* (0.104)	-0.179 (0.103)	-0.192 (0.130)	-0.231 (0.132)
Technical Knowledge	-0.704*** (0.179)	-0.792*** (0.195)	-0.787*** (0.230)	-0.727*** (0.230)
Generalized Collaboration	-0.631* (0.301)	-0.665* (0.318)	-0.581 (0.367)	0.894* (0.407)
Repeated Collaboration				0.738 (0.426)
Identity Specialization		-0.550* (0.245)	-0.657** (0.252)	-0.410*** (0.147)
Identity Specialization * Cumulated Experience			0.861* (0.374)	1.250 (0.969)
Identity Specialization * Repeated Collaboration				0.912** (0.287)
Log likelihood	-220.393	-219.313	-216.520	-215.933
Wald test (d.f)	32.30(5)	35.31 (6)	39.44 (7)	50.49(9)
N. of spells	1425	1425	1425	1425
N. of events	53	53	53	53
*p<0.05;**p<0.01;***p< 0.001 Robust standard errors are in parentheses				

Model 1 is the baseline model with control variables available for the full sample. Model 2 adds the main independent variable of theoretical interest, *Identity Specialization*. Model 3 and Model 4 include the linear interaction terms of *Experience* and *Repeated collaboration* with *Identity specialization*. As was the case in the previous chapter, I interpret the results by exponentiating the estimates from the Cox regression in order to produce hazard ratios. Analogous to odds ratios, hazard ratios with a value greater than one indicate an increase in the likelihood that an event will occur, whereas a value less than one decreases the hazard of it occurring.

Model 1 reports parameter estimates for my control variables. A one-standard-deviation increase in *Technical Knowledge* – that is, in the number of known languages – decreases on average the likelihood of dropping out by 51percent [1-exp(-0.704], possibly indicating that contributors with advanced skills have the necessary capabilities to deal with a larger and more challenging set of problems, and are better valued by other project members. Contributors who are also active in other projects have a higher chance of sustaining their participation in the focal project, suggesting that commitment increases as involvement in the whole open source community increases. The negative and significant parameter for *Generalized Collaboration* shows that, ceteris paribus, collaborative practices tend to help socialization of contributors in the community, which in turn increases on average the likelihood of sustained participation. In fact, a standard deviation increase in the *Generalized Collaboration* parameter decreases the chances of leaving the project on average by 47percent [1-exp(-0.63)]. Interestingly, this result, together with the non-significant parameter for *Cumulated Experience*, indicates that individual work activities that are not embedded specifically in social practices of interaction and collaboration do not help to foster enduring participation in open productions. Finally, the highly positive and significant parameter for *Scope of Contributions* shows that a dispersed attention spread across too many modules is detrimental to the motivation to sustain participation in the long term.

This last result strongly hints at the expected results from Model 2, in which the main independent variable of theoretical interest, *Identity Specialization*, is added to test the substantive validity of Hypothesis 1. In line with the literature on the sociology of categorization (Zuckerman, 1999; Hsu, 2006; Hannan, 2010) I find that an increase in the degree of specialization, indicating the construction of a focused identity, increases the expected tenure of contributors in open productions (i.e., decreases on average the hazard of individuals dropping out of the project). Thus the negative and significant parameter for *Identity Specialization*, taken together with the positive parameter for *Scope of Contributions*, supports Hypothesis 1. A standard deviation increase in *Identity Specialization* decreases the likelihood of leaving the project by 42 percent [1-exp(-0.55)].

Models 3 and 4 introduce some contingency factors to circumstantiate the conditions under which specialization promotes sustained participation. Hypotheses 2a and 2b are tested in Model 3. I interact *Cumulated Experience* with *Identity Specialization* with the expectation that experienced individuals benefit more from a complex than from a focused identity when they are already socialized in the community. The positive and significant parameter of the interaction term indicates that the negative effect of *Identity Specialization* on the hazard of leaving the project decreases as the level of *Cumulated Experience* increases. That is, high specialization appears beneficial for sustained participation only at lower levels of cumulated experience. To investigate further the contingent effect of experience on the relationship between specialization and sustained participation I decided to plot the dependent variable at different levels of the independent and moderating variables. In the case of event history models the typical approach is to calculate predicted values of the hazard rate under different conditions of the independent variable and moderator and showing the predicted levels of the hazard rate multiplier at these different levels of the moderator. Figure 4.2 plots the result of this exercise. The multiplier of the hazard rate is plotted on the vertical axis, while the horizontal axis represents different levels of Identity Specialization, ranging from one standard deviation below the mean to one standard deviation above the mean. As

before, values above one increase the baseline hazard of dropping out of the project, whilst values below one decrease it. The plot in Figure 4.2 shows clearly that two very different shapes of the hazard rate exist at different levels of the moderator. Ceteris paribus, the risk of exit from the project is decreased for more experienced individuals with a more spread out, complex identity, while the opposite holds true for those with low levels of experience. This argument is in line with existing research on typecasting (Faulkner, 1980; Zuckerman et al., 2003) which shows the detrimental effect of being typecast on future career opportunities once musicians and actors have already gained sufficiently long tenure in the industry.

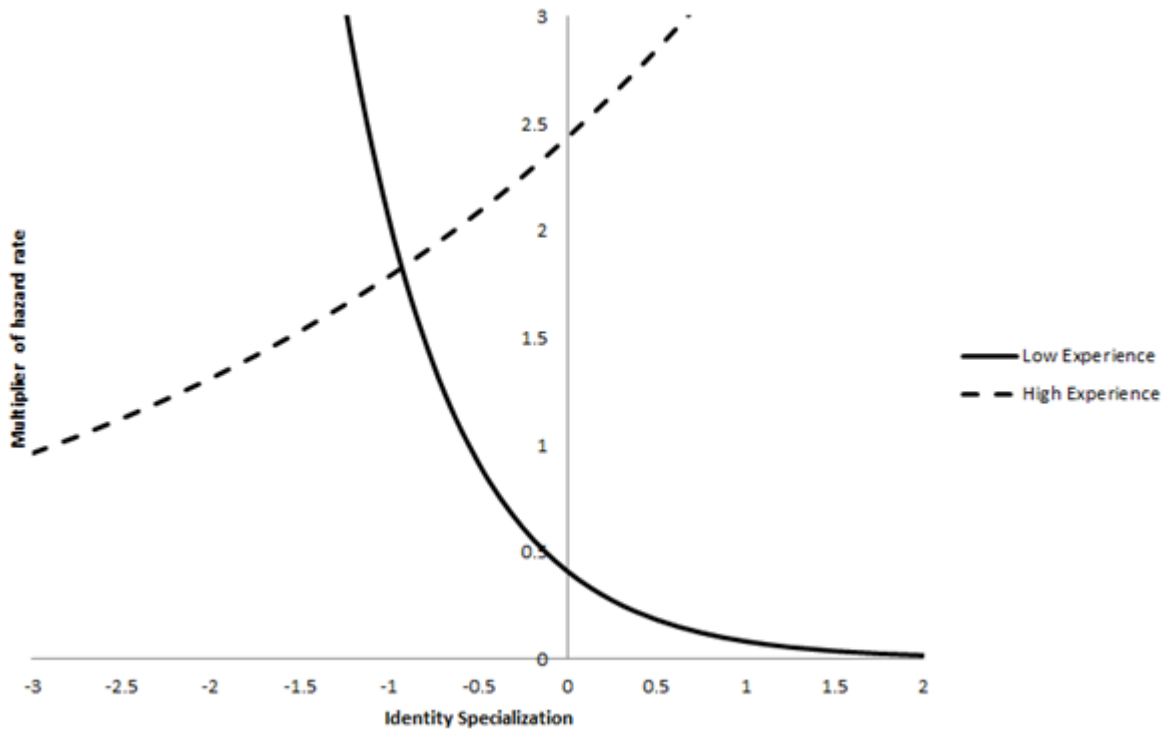


Figure 4.4: Interaction between Cumulated Experience and Identity Specialization

However, the results change as I introduce the second interaction term in Model 4. Hypotheses 3a and 3b posit that the effect of specialization on the hazard of leaving the project is contingent on the degree of prior repeated collaborations engaged in by the focal individual, such

that at higher degree of prior repeated collaboration participants with broader, more generalist identities incur a lower risk of leaving than specialists. To test these hypotheses I included in the model a second interaction term, *Identity Specialization * Repeated Collaboration*. The parameter is positive and significant, indicating that at higher levels of repeated collaboration the negative effect of specialization on the hazard of dropping out decreases. Highly collaborative contributors gain legitimation through socialization with other contributors, therefore they don't need to be constrained to a specialist profile. Figure 4.3 shows the plot of this interaction term.

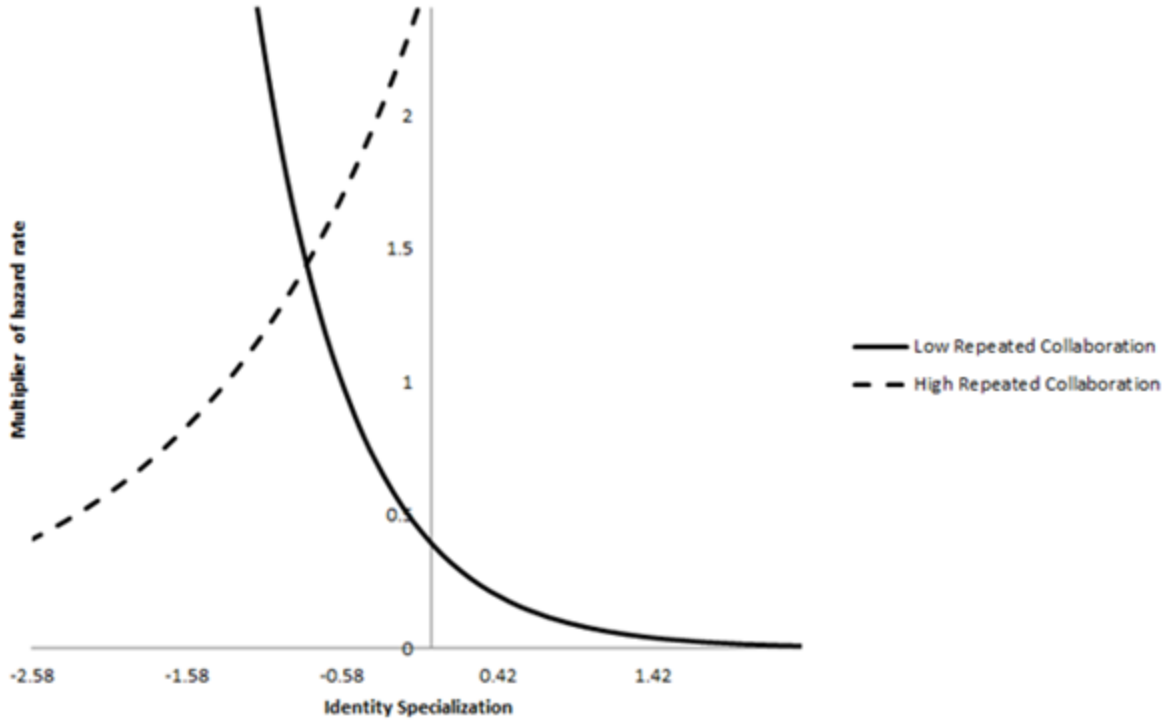


Figure 4.5: Interaction between Repeated Collaboration and Identity Specialization

However, once I include *Repeated Collaboration* in the model the interaction term with the first moderator (i.e., *Identity Specialization***Cumulated Experience*) becomes non-significant. This result has interesting implications for theories of participation in open productions and the social outcomes of categorical and typecasting processes. Simply gaining more experience is not sufficient

to guarantee effective socialization in an open production. Socialization, and consequently sustained participation, is achieved only through consistent collaborative practices. A focused identity is important for obtaining recognition in the early stages of participation. However, subsequent extensive collaborative practices requiring interaction with peers allows individuals to gain a level of legitimation that goes beyond that achieved by focusing and specializing one's expertise. Thus, once a sufficient level of collaboration is achieved, participants with a broader, more generalist identity have a higher chance of sustaining participation in the project over the longer term. These results support hypotheses 3a and 3b and question the validity of hypotheses 2a and 2b once collaborative practices are accounted for.

4.6 Discussion and Conclusions

The aim of this work was to address the following overarching question: What mechanisms sustain long-term voluntary participation within open productions? I have argued that individual work practices contribute to the construction of participants' identities and that these different identity profiles influence the way in which participants are evaluated and socialized in the project. Using data from code contributions to the Apache HTTP Server open source project, my results show that contributors with a focused, specialist identity are less likely to make an early exit from the project than contributors with a broader, more generalist identity, a tendency in line with recent theories on the sociology of categorization processes (Zuckerman, 1999; Hsu, 2006; Leung & Sharkey, 2014). However, this effect is contingent on the way in which individuals engage in collaborative practices with fellow participants. Repeated collaboration serves as a mechanism by which participants may gain legitimacy as contributors to the project without the need to confine themselves to a narrow range of problems and the specialist identity profile this confers. The underlying assumption here is that both specialist and generalist identity profiles entail costs and benefits. Specialist members enjoy certain advantages derived from having a more clearly defined

identity in the eyes of their audience, whereas generalist members are more likely to be devalued or ignored. However, generalist members, whose identity is not constrained by clear expectations from their audience, may enjoy greater role flexibility and freedom of action and may therefore achieve a longer tenure in the project (Zuckerman et al., 2003). My results address this trade-off by showing that as the extent of repeated collaboration engaged in by an individual project contributor increases, the positive effect of a specialist identity on the probability of that individual sustaining participation progressively decreases, a such that a less constraining, generalist identity becomes more favourable to ongoing involvement. Repeated collaboration thus enables contributors to gain the necessary level of socialization and legitimation in the project to allow them to broaden their identity profile without incurring a cost to their reputation in the form of a devaluation of their expertise by other participants.

My study contributes to two main areas of interest. Firstly, it adds to the literature on open productions, and in particular to the stream of research addressing the issue of sustained participation. While previous research has devoted a great deal of attention to the way in which contributors in open productions are intrinsically and extrinsically motivated, ex-ante, to dedicate their time and contributions to the creation of a public good (Lakhani & von Hippel, 2003; Roberts et al., 2006; von Krogh et al., 2003; 2012), this study shifts attention to the nature of these contributions over time and to the consequences this activity has for the sustainability of individual participation. I extend recent work by Fang and colleagues (Fang & Neufeld, 2009; Qureshi & Fang, 2011; Sun, Fang & Lim, 2012) which considered participation, not merely as a by-product of initial motivation, but rather as a dynamic process that unfolds over time as a result of an individual's activities. This paper provides insights into the specific nature of these activities (i.e., identity construction, repeated collaboration) and demonstrates that sustained participation is a natural outcome emerging from successful socialization processes between contributors and their communities. This finding also extends earlier work by von Krogh et al. (2003) and Shah (2006),

which demonstrated that short-term and long-term contributors to open productions are motivated by different factors, the former by idiosyncratic needs and the latter by enjoyment and identification with the community's values. I extend this work by suggesting that no matter what the initial motivation, collaborative practices undertaken over time shape contributor identity and the chances of long-term, successful participation within the community.

Secondly, this work contributes to the literature on the sociology of categorization processes and typecasting by identifying a contingent factor influencing the likelihood of success of those with specialist versus generalist identity profiles. The very influential paper by Zuckerman and colleagues on typecasting (2003) argued that tenure in the film industry is a very important factor in determining the dynamics of actor typecasting. They demonstrated that the benefit of having a focused identity is much greater for novice actors, in the process of building their reputation, whilst actors with longer tenure, who have already gained recognition within the industry, enjoy greater flexibility to be cast in a broader spectrum of genres. My work extends this view by arguing that participants in professional settings interact with their peers by means of their work activities. Instances of interaction and collaboration create the necessary conditions for organizational participants to be successfully integrated and become legitimate members of the community. My results suggest that experience is not a sufficient explanation for this process, once I control for repeated collaboration. Organizational participants do not become legitimate members of the community simply by accumulating experience; they also need to repeatedly collaborate with other participants. Having shored up their reputation within the community in this way, experienced collaborators can then afford to extend their work into other areas of specialism, building a more generalist identity profile without incurring the risk of reputation devaluation.

This study has a number of limitations that may open up avenues for further research. Firstly, although the Apache Software Foundation has openly stated in its guidelines that all contributors are voluntary participants and are not paid by the foundation, there exist a few hybrid

situations where some contributors work for third party companies interested in developing Apache. However, at present public data on employment situations are scattered online and information on extrinsic motivations is very difficult to gather, especially for early contributors who have not been involved in the project for very long. Secondly, although this work relies on constructs such as socialization and legitimation, they are not explicit. These mechanisms are implied by the fact that participants who contribute for a fairly long period of time (i.e., approximately six months) have to be nominated and successfully elected in order to gain write access to the repository and voting rights. However, future research could address this issue and explicate the way in which role advancements foster sustained participation. Thirdly, this paper does not account for differences in performance between contributors, in terms of acceptance or rejection of modifications. Future research could delve into this topic, mining performance data within developer mailing lists to reconstruct the complete history of modification requests in order to investigate the performance feedback mechanisms that influence sustained participation.

References

Adamic, L. A., Wei, X., Yang, J., Gerrish, S., Nam, K. K., & Clarkson, G. S. (2010). *Individual focus and knowledge contribution*. Working paper.

Aiken, L. S., & West, S. G. (1991). *Multiple regression: Testing and interpreting interactions*. Sage.

Alvesson, M., & Willmott, H. (2002). Identity regulation as organizational control: Producing the appropriate individual. *Journal of Management Studies, 39*(5), 619–644.

Anderson, K. (2012). Specialists and Generalists: Equilibrium Skill Acquisition Decisions in Problem-solving Populations. *Journal of Economic Behavior in Organizations, 84*(1)

Baldwin, C. Y. (2008). Where do transactions come from? Modularity, transactions, and the boundaries of firms. *Industrial and corporate change, 17*(1), 155-195.

Baldwin, C. Y., & Clark, K. B. (2000). *Design rules: The power of modularity*. MIT press.

Baldwin, C. Y., & Clark, K. B. (2006). The architecture of participation: Does code architecture mitigate free riding in the open source development model?. *Management Science, 52*(7), 1116-1127.

Baldwin, C., & von Hippel, E. (2011). Modeling a paradigm shift: From producer innovation to

user and open collaborative innovation. *Organization Science, 22*(6), 1399-1417.

Becker, H. S. (1973). Labelling Theory Reconsidered. In *Outsiders: Studies in the Sociology of Deviance*, pp. 177–208. New York: Free Press.

Cohen, M. D., March, J. G., & Olsen, J. P. (1972). A garbage can model of organizational choice. *Administrative science quarterly, 1*(25).

Cox, D. R. (1972). Regression models and life-tables (with discussion). *Journal of the Royal Statistical Society, Series B, 34*(2), 187-220.

Crowston, K., Annabi, H., & Howison, J. (2003). Defining open source software project success, in *Proceedings of the 24th International Conference on Information Systems*, Seattle WA.

Crowston, K., & Scozzi, B. (2008). Bug fixing practices within free/libre open source software development teams. *Journal of Database Management (JDM), 19*(2), 1-30.

Crowston, K., Wei, K., Howison, J., & Wiggins, A. (2012). Free/Libre open-source software development: What we know and what we do not know. *ACM Computing Surveys (CSUR), 44*(2), 7.

Dahlander, L., & O'Mahony, S. (2011). Progressing to the center: Coordinating project work. *Organization Science, 22*(4), 961-979.

Deci, E. L., & Ryan, R. M. (2000). The" what" and" why" of goal pursuits: Human needs and the self-determination of behavior. *Psychological inquiry, 11*(4), 227-268.

Ducheneaut, N. (2005). Socialization in an open source software community: A socio-technical analysis. *Computer Supported Cooperative Work (CSCW), 14*(4), 323-368.

Fang, Y., & Neufeld, D. (2009). Understanding sustained participation in open source software projects. *Journal of Management Information Systems*, *25*(4), 9-50.

Faulkner, R. R. (1983). *Music on demand.* Transaction Publishers.

Fielding, R. T. (1999). Shared leadership in the Apache project. *Communications of the ACM, 42*(4), 42-43.

Franke, N., & Hippel, E. V. (2003). Satisfying heterogeneous user needs via innovation toolkits: the case of Apache security software. Research Policy, 32(7), 1199-1215.Ghosh, R. A. (1998). Interview with Linus Torvalds: What motivates free software developers? *First Monday 3*(3).

González-Barahona, J. M., López, L., & Robles, G. (2004). Community structure of modules in the apache project. In *Proceedings of the 4th Workshop on Open Source Software Engineering*. 26th International Conference on Software Engineering, Edinburgh, Scotland, UK.

Handley, K., Sturdy, A., Fincham, R., and Clark, T. (2006) Within and beyond communities of practice: Making sense of learning through participation, identity and practice. *Journal of Management Studies, 43*(3), 641–653.

Hann, I. H., Roberts, J., Slaughter, S. A., & Fielding, R. (2002). Economic incentives for participating in open source software projects. *Proceedings of the Twenty-Third International Conference on Information Systems*, 365-372.

Hann, I. H., Roberts, J., & Slaughter, S. A. Why developers participate in open source software projects: An empirical investigation. *Twenty-Fifth International Conference on Information Systems*.

Hannan, M.T. (2010) Partiality of Memberships in Categories and Audiences. *Annual Review of Sociology, 36*: 159-181.

Hannan, M. T., Pólos, L., & Carroll, G. R. (2007). *Logics of organization theory: Audiences, codes, and ecologies.* Princeton University Press.

Hars, A., Ou, S., (2002). Working for free? Motivations for participating in Open-Source projects. *International Journal of Electronic Commerce, 6*, 25–39

Henderson, R. M., & Clark, K. B. (1990). Architectural innovation: the reconfiguration of existing product technologies and the failure of established firms. *Administrative science quarterly*, 9-30.

Henkel, J., & Baldwin, C. Y. (2009). *Modularity for value appropriation: Drawing the boundaries of intellectual property.* Harvard Business School.

Hertel, G., Niedner, S., and Herrmann, S. (2003). Motivation of software developers in open source projects: An Internet-based survey of contributors to the Linux kernel. *Research Policy, 32*(7) , 1159–1177.

Howison, J., Inoue, K., & Crowston, K. (2006). Social dynamics of free and open source team communications. In *Open Source Systems*, 319-330.

Hsu, G. (2006). Jacks of all trades and masters of none: Audiences' reactions to spanning genres in feature film production. *Administrative science quarterly, 51*(3), 420-450.

Krishnamurthy, S. (2002). Cave or community?: An empirical examination of 100 mature open source projects. *First Monday*.

Lakhani, K. R., & Von Hippel, E. (2003). How open source software works:"free" user-to-user assistance. *Research policy, 32*(6), 923-943.

Langlois, R. N., & Garzarelli, G. (2008). Of Hackers and Hairdressers: Modularity and the Organizational Economics of Open-source Collaboration. *Industry and Innovation, 15*(2), 125-143.

Lave, J., & Wenger. (1990) *Situated Learning. Legitimate Peripheral Participation*. Cambridge: Cambridge University Press.

Lerner, J., & Tirole, J. (2001). The open source movement: Key research questions. *European Economic Review, 45*(4), 819-826.

Lerner, J., & Tirole, J. (2002). Some simple economics of open source. *The journal of industrial economics, 50*(2), 197-234.

Leung, M. D., & Sharkey, A. J. (2014). Out of sight, out of mind? Evidence of perceptual factors in the multiple-category discount. *Organization Science, 25*(1), 171-184.

Levine, S. S., & Prietula, M. J. (2014). Open Collaboration for Innovation: Principles and Performance. *Organization Science*. In press.

Lomi, A., Conaldi, G., & Tonellato, M. (2012). Organized anarchies and the network dynamics of decision opportunities in an open source software project. *Research in the Sociology of Organizations, 36*, 363-397.

Markus, L., Manvile, B., & Agres, C. (2000). What makes a virtual organization work? *MIT Sloan Management Review 42*(1), 13–26.

Negro, G., & Leung, M. D. (2013). Actual and perceptual effects of category spanning. *Organization Science, 24*(3), 684-696.

Phillips, D. J., & Zuckerman, E. W. (2001). Middle-Status Conformity: Theoretical Restatement and Empirical Demonstration in Two Markets1. *American Journal of Sociology, 107*(2), 379-429.

Raymond, E. (1999). The cathedral and the bazaar. *Knowledge, Technology & Policy, 12*(3), 23-49.

Roberts, J. A., Hann, I. H., & Slaughter, S. A. (2006). Understanding the motivations, participation, and performance of open source software developers: A longitudinal study of the Apache projects. *Management science, 52*(7), 984-999.

Robles, G., Koch, S., Gonzalez-Barahona, J. M., & Carlos, J. (2004). Remote analysis and measurement of libre software systems by means of the CVSAnalY tool. *In Proceedings of the 2nd ICSE Workshop on Remote Analysis and Measurement of Software Systems (RAMSS),*. 51-55.

Robles, G. & Gonzalez-Barahona, J. M. (2005). Evolution of volunteer participation in libre software projects: Evidence from Debian. *Proceedings of the First International Conference on Open Source Systems*, 100-107.

Scacchi, W. (2002, February). Understanding the requirements for developing open source software systems. *In Software IEE Proceedings* (Vol. 149, No. 1, pp. 24-39). IET.

Shah, S. K. (2006). Motivation, governance, and the viability of hybrid forms in open source software development. *Management Science, 52*(7), 1000-1014.

Simon, H. A. (1965). The architecture of complexity. *General systems, 10*, 63-76.

Von Krogh, G., Spaeth, S., & Lakhani, K. R. (2003). Community, joining, and specialization in open source software innovation: a case study. *Research Policy, 32*(7), 1217-1241.

Von Krogh, G., Haefliger, S., Spaeth, S., & Wallin, M. W. (2012). Carrots and rainbows: Motivation and social practice in open source software development. *MIS Quarterly, 36*(2), 649-676.

Zerubavel, E. (1997). *Social Mindscapes: An Invitation to Cognitive Sociology*. Harvard University Press, Cambridge, MA.

Zuckerman, E. (1999) The Categorical Imperative: Securities Analysts and the Illegitimacy Discount. *American Journal of Sociology 104*(5) 1398-1438.

Zuckerman, E. W., Kim, T. Y., Ukanwa, K., & von Rittmann, J. (2003). Robust Identities or Nonentities? Typecasting in the Feature-Film Labor Market. *American Journal of Sociology, 108*(5), 1018-1073.

Chapter 5

CONCLUSIONS

5.1 Summaries of chapter results and contributions

The overarching goal of this dissertation was to provide a framework for the concept of "open production", examining its internal organizational logic and showing how it emerges from, and extends beyond, the basic model of problem-solving organizations. To do this I considered open source software projects as specific solutions to the problem of economic production. As such, open productions face the same issues as conventional hierarchical organizations, such as division of labor, collaboration, and coordination. I have built on previous work on open innovation (see Lakhani, Lifshitz-Assaf & Tushman, 2013) but examined a wider range of processes, encompassing several foundational concepts in organizational sociology and cognition: attention allocation (Chapter 2), group learning and expertise diversity (Chapter 3), identity construction and sustained participation (Chapter 4). My work complements current explanations based on motivation and incentives to show how open productions are sustained through mundane work practices that create complex interdependencies between problems and problem-solvers. I have provided examples in which both individual decision-making and organizational outcomes are embedded in, and determined by, a transparent problem-solving structure. In particular, I have illustrated three fundamental processes that served as dependent variables in my studies and which formed the basis for my research questions within the context of self-managing teams with fluid boundaries. These were: i) problem selection, ii) problem resolution, and iii) sustained participation.

Building on existing research, my work adds a focus on the micro-organizational mechanisms that make open production effectively possible as a strategy for technology development. Unlike much of the prior literature on open productions, I do not emphasize motivation, status-seeking or efficient allocation of property rights. I concentrate instead on the actual work practices– the day-to-day activities that are involved in open productions and ultimately make them sustainable problem-solving arrangements. I focus on individual acts of problem solving– the smallest possible constituent unit of work practices. As such, the research design I

have implemented and the models I have developed have allowed me to examine work practices at a level of detail that has so far been inaccessible for prior research.

In the following sections I summarize the main findings for each of the three empirical chapters and discuss their contributions to the literature on organization theory. I then identify the scope conditions for my studies and derive implications for the generalizability of the arguments advanced in the dissertation.

5.1.1 Chapter 2

In this chapter I investigated the dynamics of attentional processes and the importance of their role in problem-solving and decision-making in open productions. Drawing from the perspective of the Carnegie School in which problems compete for the limited attention of participants (March & Simon, 1958), I presented two attentional mechanisms – attention clustering and attention spread – that guide participants' decisions regarding how to allocate attention among problems. I showed that apparently individualistic attention allocation decisions are in fact the result of the embeddedness of individuals in a complex network of interdependencies linking problems and problem-solvers in an organization. When deciding which problems to engage with participants do not rely solely on those which fall immediately within the range of their own attention; they also take into consideration problems to which neighboring individuals have turned their attention.

The analysis shows a positive tendency towards attention clustering, indicating that attentional processes tend to follow repetitive patterns of joint interest towards problem-solving: collaborating on a problem makes it more likely that two contributors will allocate their joint attention to future problems. Furthermore the analysis suggests that problems tend to be recipients of a disassortative attention spread, indicating that already popular software bugs are less likely than their less popular counterparts to attract additional attention in the future from highly active contributors. As a consequence of the associated tendency toward decentralized problem-solving,

attention is more homogeneously distributed across problems, an important contributory factor for the viability of the project, since the entire portfolio of presenting problems requires resolution, both those that are ‘attention-grabbing’ and the more obscure.

This chapter contributes to the literature on organizational attention by extending the current understanding of attentional processes as either purely bottom-up or top-down (Ocasio, 2011). I propose that attention is ultimately a function of other members' attention, giving rise to attention networks, where transparent attentional cues are used by individuals to assess and select objects of attention for problem-solving activity. This paper also extends recent work on distributed social cognition (Smith & Collins, 2009; Kaplan, 2011) which showed that market evaluations don’t occur in a vacuum, but are embedded in attention networks linking interdependent actors’ views (Prato & Stark, 2013). What I have added to this perspective is a series of specific mechanisms through which individual acts of attention allocation concatenate and reproduce behavioral patterns that drive collective problem-solving efforts.

5.1.2 Chapter 3

In this paper I examined the contested relationship between group members' repeated collaboration and group performance in the context of bug-fixing in an open source software project. I took as my point of departure competing theories and evidence on the effects of repeated collaboration, showing both positive and negative potential impact on group performance. I then sought to delineate more precisely the boundary conditions that determine the likely direction of the effect. In particular, I investigated the role of the distribution of task-related expertise among group members as a determining contingent factor in this dynamic relationship. My analysis shows that the relationship between group experience of working together and group performance is non-monotonic, with a state of decreasing returns coming into force beyond a certain point. I also show that expertise diversity moderates the relationship between repeated collaboration and team

performance. Groups composed of individuals with homogeneous expertise perform better than groups with heterogeneous expertise at lower levels of prior collaboration, because homogeneity helps establish common ground in the early stages of a group project (Kraut et al., 2002). Whereas at later stages of a project, where groups have a longer track-record of collaboration, those that are composed of individuals with heterogeneous expertise perform better than those with homogeneous expertise. In this latter scenario specialization and expertise diversity are considered critical to the maintenance of group performance in the long run (Kane, Argote & Levine, 2005). My investigation contributes to two major areas of theoretical interests. Firstly, this work extends theory on organizational learning by investigating contingent factors that unveil the way in which performance-based learning is achieved in open productions. The importance of a contingent learning context is underlined by the moderating effect of expertise diversity shown here, which contrasts with research in other settings, substantiating arguments about unconditional and linear effects of experience of working together (Boh et al., 2007). Existing research (Reagans et al., 2005) has laid out a theoretical case for the value of specialized task-related expertise as an aid to the development of transactive memory (theorizing the role of expertise diversity), and about the value of common ground, both to allow effective coordination across functions and knowledge–sharing (theory regarding repeated collaboration). However, because these two phenomena have advantages and disadvantages that offset each other, they are frequently confounded in empirical analysis. Distinguishing the two, as was made possible in my analysis, is significant from both an empirical and a theoretical standpoint. By specifying further the interaction effect between these two constructs we are able to advance our understanding about the kind of conditions under which it is preferable to have a diverse versus homogeneous work group.

The second key contribution of my study is to the body of theoretical and empirical work on open productions, advancing understanding of their distinct features. Amongst these the dynamics of collaboration emerge as an important area for investigation. As such, this study sought to

illuminate the particular causes that influence the way in which expertise diversity interacts with repeated collaboration, the latter being a process which has already presented itself as a central issue in the emerging literature on open production and project-based organizations (Sorenson & Waguespack, 2006; Perretti & Negro, 2007; Bakker, 2010).

5.1.3 Chapter 4

In Chapter 4 I argued that the identities of individual project participants, as repositories of project-relevant expertise, emerge from the substance of the contributions they make through their ongoing work, and that these different identity profiles influence the likelihood that participants will sustain participation in an open production. Using data from code contributions to the Apache HTTP Server open source project, my results show that, in line with recent theories on the sociology of categorization processes (Zuckerman, 1999; Hsu, 2006; Leung & Sharkey, 2014), contributors with a focused, specialist identity are less likely to make an early exit from a project than contributors with a broader, more generalist identity, However, this effect is shown to be contingent on the way in which individuals engage in collaborative practices with fellow participants. Repeated collaboration serves as moderating mechanism, whereby legitimacy as a valid and valuable contributor to the project may be obtained without the need to demonstrate narrow, specialist expertise. As such, repeat-collaborating generalists gain an advantage over specialists, in terms of the probability of sustaining participation in a project over time.

This study contributes to two main areas of interest. Firstly, it adds to the literature on open productions with respect to the stream of research addressing the issue of sustained participation. It achieves this by shifting attention from the motivations of contributors to commence participation in a project to the nature of these contributions over time and the consequences of this activity for the sustainability of individual participation. I extend recent work by Fang and colleagues (Fang & Neufeld, 2009; Qureshi & Fang, 2011; Sun, Fang & Lim, 2012) which considered participation, not

merely as a by-product of initial motivation, but rather as a dynamic process that unfolds over time as a result of an individual's activities. In this paper the constituent characteristics of these activities (i.e., identity construction, repeated collaboration) are elucidated, demonstrating that sustained participation is a natural outcome emerging from successful socialization processes between contributors and their communities. Earlier work by Von Krogh et al. (2003) and Shah (2006) found that short-term and long-term contributors to open productions are motivated by different factors, the former by idiosyncratic needs and the latter by enjoyment and identification with the community's values. My study extends this work by suggesting that, regardless of the initial motivation for participation, it is the engagement in successive collaborative efforts over time that helps shape the identity of contributors within the community, and that this is a key explanatory factor in their on-going participation in the longer-term.

A second key area to which this study contributes is the literature on the sociology of categorization processes and typecasting. This is accomplished by identifying a contingent factor influencing the likelihood of success of those with specialist versus generalist identity profiles. My work builds on the work of Zuckerman et al. (2003) by arguing that participants in professional settings interact with their peers by means of their work activities. Organizational participants become successfully integrated, legitimate members of the community through instances of interaction and collaboration. My results suggest that, once repeated collaboration is controlled for, experience alone is not a sufficient explanation for this process. In other words, organizational participants do not become legitimate members of the community simply by accumulating experience; they must also engage in ongoing collaboration with other participants. Having thus consolidated their reputation within the community, experienced collaborators can afford to extend their work into other areas of specialism, constructing a broader, more generalist identity profile, without incurring the risk of reputation devaluation.

5.2 Scope conditions and limitations

Here I discuss the scope conditions for the results reported in chapters 2 to 4, beyond the specific limitations and suggestions for further research that I identified within those chapters. These conditions have important implications for the generalizability of my findings. My hypotheses were tested in the context of F/OSS projects, which have been conceptualized as private-collective innovation models combining elements of individual investment and collective action (von Hippel & von Krogh, 2003). This economic model is based on teams of geographically distributed participants who often work on a voluntary basis and are motivated to invest their time and effort to create non-rival and non-excludable public goods innovations. This private-collective model of collaboration, innovation and production is however generalizable beyond software. In the introduction to this dissertation I introduced the term "open production" to illustrate production ecosystems that (i) create goods of economic value; (ii) grant open access to participants to contribute and consume freely; (iii) are based on constant interactions and information exchange; (iv) purposefully coordinate participants' labor. This phenomenon has seen a rapid proliferation in recent years, to which organization and innovation researchers have responded with growing interest, as the literature on user and open innovation indicates (for a review, see Levine & Prietula, 2014; Baldwin & von Hippel, 2011). Examples of open productions can typically be found in user innovation communities, whereby organizations reach out to a community of users who contribute voluntarily to the development and design of innovative products and services (Jeppesen & Frederiksen, 2006). These communities exist across a variety of industries, including sports equipment (Franke & Shah, 2003; Lüthje, 2004), the automotive sector (Ili, Albers & Miller, 2010), biotechnologies (Bianchi et al., 2011), musical instruments (Jeppesen & Frederiksen, 2006), and software development (Dahlander, Frederiksen & Rullani, 2008). All open production systems can be very heterogeneous in terms of incentives structure and organizational design. However in my research I identified three foundational organizational features – i.e., modularity, transparency and

technology-mediated communication – that open productions must possess in order to be identified as such. I argue that these features help us identify the scope conditions within which my results can be generalized.

5.2.1 Modularity

A production system is assumed to be modular when its parts are capable of functioning independently but are also able to operate in conjunction to support the whole. A modular system is one in which the "constituent parts – resources, tasks, or components – are partitioned into subsets called modules" (Baldwin & von Hippel, 2011: 1401). Modularity refers to what Herbert Simon called near-decomposability (Simon, 1965). Elements contained within each module show a high degree of interdependence, in the sense that modifications to one element are tightly coupled with modifications to other elements. However elements spread across modules show a high degree of independence, in the sense that modifications to one element are very loosely coupled with modifications to other elements (Thompson, 1967; Baldwin & Clark, 2000). Modularity is critical for collaborative systems such as open productions because production activities can be carried out for each module in parallel, without requiring a high degree of integration and communication between modules. Contributors active in separate modules that belong to a larger system do not need to work simultaneously or to be colocated. They can set up an infrastructure for asynchronous work that will integrate the single modules together and make up a system that functions as a whole. As I showed in Chapter 4, modular systems are crucial for inducing an effective division of labor and specialization of skills within complex open productions. In more traditional settings these are achieved through formal hierarchies and contracts conceived to overcome differences in principal-agent incentives. Production systems that do not exhibit a high degree of modularity face a much greater struggle to create the knowledge boundaries necessary for a more efficient division of labor, a process upon which effective problem-solving and task self-assignment processes are based.

5.2.2 Transparency

In open production systems or within system modules, participants can rely on social transparency rather than modularity to achieve coordination. Social transparency refers to the fact that each contributor's activities are "transparent" to other participants. Most work practices are carried out using social applications that let members track and follow the activities of other members, irrespective of their location. Every participant works separately to improve the system, contributing upon the transparent work of other participants. In open collaborative projects, transparency and modularity usually build on each other, both contributing to coordination and division of labor (Colfer & Baldwin, 2010). As I've shown in this dissertation, in open source software this new approach mixes version control systems with features of social media to create transparent work environments, where every action undertaken by any individual is immediately visible and traceable by other project members (Dabbish et al., 2013). Contributors to open productions keep everyone up to date on things they do or work on and in turn decide which individuals or problems of interest they wish to pay attention to. Since social transparency allows participant to be aware of what feature of the project is being modified, when, where and by whom, this meta-information is used by other participants to coordinate their efforts and respond to changes in content appropriately (Stuart et al., 2012). In Chapter 2 I argued that these kinds of attentional cues will produce direct and indirect effects on the attention allocation mechanisms of other members, and their consequent decisions about self-assignment to problems. In Chapters 3 and 4 I argued that transparent environments – coupled with effective communication structures – are necessary for the establishment of group processes, such as recognition of other participants' expertise and the formation of Transactive Memory Systems (TMS). In production systems that are not based on a transparent environment – in which each action is visible and traceable – participants lack the means to gauge the expertise of other contributors and cannot form a clear idea of the distribution of skills within the community. Furthermore, a lack of transparent infrastructure

inhibits coordination as it makes it more difficult for contributors to respond efficiently to changes in content by other contributors.

5.2.3 Technology-mediated communication

Communication is a foundational feature of open productions, which are by definition characterized by geographically distributed teams. Contributors use communication channels to learn about and evaluate other contributors (Rulke & Rau, 2000), tap into otherwise inaccessible knowledge pools (Palazzolo, 2005) and develop tacit knowledge that eventually leads to the creation of working routines (Hollingshead & Brandon, 2003). In this dissertation I modelled problem-solving and collaborative activities within the context of geographically dispersed software development groups that rely mostly on instant messaging and email for day-to-day communication, as these asynchronous media are relatively cheap and easy to use. Distributed groups may also communicate in other ways, such as by video or audio conference, but they are unlikely to have as much face-to-face interaction as a typical co-located group. When group members are separated from one another geographically they can experience decreased cohesion within their work group. Nevertheless, although their development may take longer, there is evidence that the group cognition processes on which I base my arguments – such as TMS (Chapter 3) and identity evaluation (Chapter 4) – can develop effectively in groups that collaborate solely through computer-mediated communication (Kanawattanachai & Yoo, 2007). It is customary for open productions to create dedicated mailing lists and an instant messaging infrastructure to stimulate discussions around every single change to the product or service under development. Transparent discussions and private communication are the biggest information source for participants who learn about the community and its contributors. Production systems that fail to sustain the centrality of communication channels are at risk of being populated by segregated participants who lack any means to integrate their knowledge with the rest of the community.

The three features discussed above constitute the backbone of open production that exploits the distributed knowledge of geographically dispersed volunteers who dedicate their time and effort to the creation of public goods. These features also define the scope conditions within which my findings are generalizable relative to foundational organizational aspects of open productions.

As we have seen open productions represent a relatively new attempt to achieve technological innovation through an alternative use of incentive structures and intellectual property strategy. Contributors have the right to access and modify the product design and a responsibility to guarantee that every modification to the product design is accessible by other contributors. In this sense it seems we are witnessing the rise and rapid diffusion of a new form of economic production, and with it the demise of the traditional separation between the roles of producer and consumer. In this environment the consumer has been, to varying degrees, integrated into the role of producer and is thus largely responsible for the innovation of products and services. The implications of this boundary erosion – such as the reframing of the role of markets as interfaces where supply and demand meet – are far-reaching and, as such, they are beyond the scope of this dissertation. They do, however, represent fascinating avenues for future research. The studies I have presented here provide evidence of organizational mechanisms operating specifically within the F/OSS context that nevertheless speak coherently to existing organizational and sociological theory. Whilst remaining confined in direct generalizability to environments exhibiting the core features outlined in this chapter, I therefore hope I may contribute in some way to inspiring future research beyond the scope of these studies.

References

Bakker, R. M. (2010). Taking stock of temporary organizational forms: A systematic review and research agenda. International Journal of Management Reviews, 12(4), 466-486.

Baldwin, C. Y., & Clark, K. B. (2000). Design rules: The power of modularity. MIT press.

Baldwin, C., & von Hippel, E. (2011). Modeling a paradigm shift: From producer innovation to user and open collaborative innovation. Organization Science, 22(6), 1399-1417.

Bianchi, M., Cavaliere, A., Chiaroni, D., Frattini, F., & Chiesa, V. (2011). Organisational modes for open innovation in the bio-pharmaceutical industry: an exploratory analysis. *Technovation*

Boh, W., Slaughter, S. A., & Espinosa, J. A. (2007). Learning from experience in software development: A multilevel analysis. Management Science, 53(8), 1315-1331.

Colfer, L., & Baldwin, C. Y. (2010). The mirroring hypothesis: Theory, evidence and exceptions. Harvard Business School, 10-058.

Dabbish, L., Stuart, C., Tsay, J., & Herbsleb, J. (2012, February). Social coding in GitHub: transparency and collaboration in an open software repository. In Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work (pp. 1277-1286). ACM.

Dahlander, L., Frederiksen, L., & Rullani, F. (2008). Online communities and open innovation. *Industry and innovation, 15*(2), 115-123.

Fang, Y., & Neufeld, D. (2009). Understanding sustained participation in open source software projects. Journal of Management Information Systems, 25(4), 9-50.

Franke, N., & Shah, S. (2003). How communities support innovative activities: an exploration of assistance and sharing among end-users. Research policy, 32(1), 157-178.

Hollingshead, A. B., & Brandon, D. P. (2003). Potential benefits of communication in transactive memory systems. *Human Communication, 29*(4), 607-615.

Hsu, G. (2006). Jacks of all trades and masters of none: Audiences' reactions to spanning genres in feature film production. Administrative science quarterly, 51(3), 420-450.

Ili, S., Albers, A., & Miller, S., (2010). Open innovation in the automotive industry. *R&D Management 40*(3), 246–255.

Jeppesen, L. B., & Frederiksen, L. (2006). Why do users contribute to firm-hosted user communities? The case of computer-controlled music instruments. Organization science, 17(1), 45-63.

Kanawattanachai, P., & Yoo, Y. (2007). The impact of knowledge coordination on virtual team performance over time. *MIS Quarterly, 31*(4), 783–808.

Kane, A. A., Argote, L., & Levine, J. M. (2005). Knowledge transfer between groups via personnel rotation: Effects of social identity and knowledge quality. Organizational Behavior and Human Decision Processes, 96(1), 56-71.

Kaplan, S. 2011. Research in Cognition and Strategy: Reflections on Two Decades of Progress and a Look to the Future. Journal of Management Studies, 48: 665-695

Kraut, R. E., Fussell, S. R., Brennan, S. E., & Siegel, J. (2002). Understanding effects of proximity on collaboration: Implications for technologies to support remote collaborative work. Distributed work, 137-162.

Lakhani, K. R., Lifshitz-Assaf, H., & Tushman, M. (2013). Open innovation and organizational boundaries: task decomposition, knowledge distribution and the locus of innovation. *Handbook of economic organization: Integrating economic and organizational theory*, 355-382.

Levine, S. S., & Prietula, M. J. (2014). Open Collaboration for Innovation: Principles and

Performance. Organization Science. In press.

Leung, M. D., & Sharkey, A. J. (2014). Out of sight, out of mind? Evidence of perceptual factors in the multiple-category discount. Organization Science, 25(1), 171-184.

Lüthje, C. (2004). Characteristics of innovating users in a consumer goods field: An empirical study of sport-related product consumers. *Technovation 24*(9) 683–695.

March, J. G., H. Simon. 1958. Organizations. John Wiley & Sons, New York.

Ocasio, W. 2011. Attention to attention. Organization Science, 22(5): 1286-96.

Palazzolo, E. T. (2005). Organizing for information retrieval in transactive memory systems. *Communication Research, 32*, 726–761.

Perretti, F., & Negro, G. (2007). Mixing genres and matching people: a study in innovation and team composition in Hollywood. Journal of Organizational Behavior, 28(5), 563-586.

Prato, M., & Stark, D. (2013, January). Peripheral Vision in Financial Markets: How attention networks shape valuation. In Academy of Management Proceedings (Vol. 2013, No. 1, p. 15923). Academy of Management.

Qureshi, I., & Fang, Y. (2010). Socialization in open source software projects: A growth mixture modeling approach. Organizational Research Methods, 14(1), 208-238

Reagans, R., Argote, L., & Brooks, D. (2005). Individual experience and experience working together: Predicting learning rates from knowing who knows what and knowing how to work together. Management science, 51(6), 869-881.

Rulke, D. L., & Rau, D. (2000). Investigating the encoding process of transactive memory development in group training. *Group & Organization Management, 25*(4), 373-396.

Simon, H. A. (1965). The architecture of complexity. General systems, 10, 63-76.

Shah, S. K. (2006). Motivation, governance, and the viability of hybrid forms in open source software development. Management Science, 52(7), 1000-1014.

Smith, E. R., & Collins, E. C. (2009). Contextualizing person perception: Distributed social cognition. Psychological Review, 116: 343–364.

Sorenson, O., & Waguespack, D. M. (2006). Social structure and exchange: Self-confirming dynamics in Hollywood. Administrative Science Quarterly, 51(4), 560-589.

Stuart, H. C., Dabbish, L., Kiesler, S., Kinnaird, P., & Kang, R. (2012). Social transparency in networked information exchange: a theoretical framework. In Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work (pp. 451-460). ACM.

Sun, Y., Fang, Y., & Lim, K. H. (2012). Understanding sustained participation in transactional virtual communities. Decision Support Systems, 53(1), 12-22.

Thompson, J. (1967). *Organizations in Action: Social Science Bases of Administrative Theory*. Transaction Publishers. New Brunswick (NJ)

von Hippel, E., & von Krogh, G. (2003). Open source software and the "private-collective" innovation model: Issues for organization science. Organization science, 14(2), 209-223.

Von Krogh, G., Spaeth, S., & Lakhani, K. R. (2003). Community, joining, and specialization in open source software innovation: a case study. Research Policy, 32(7), 1217-1241.

Zuckerman, E. (1999) The Categorical Imperative: Securities Analysts and the Illegitimacy Discount. American Journal of Sociology 104(5) 1398-1438.

Zuckerman, E. W., Kim, T. Y., Ukanwa, K., & von Rittmann, J. (2003). Robust Identities or Nonentities? Typecasting in the Feature-Film Labor Market. American Journal of Sociology, 108(5), 1018-1073.